A CLOCKWORK KIDNEY

ROBERT MOSS

Using hierarchical dynamical networks to model emergent dynamics in the kidney

September 2008

Department of Computer Science and Software Engineering
School of Engineering, The University of Melbourne

Submitted in total fulfilment of the requirements of the degree of Doctor of Philosophy
Robert Moss. *A Clockwork Kidney*: Using hierarchical dynamical networks to model emergent dynamics in the kidney
© September 2008

**SUPERVISORS:**
Edmund Kazmierczak
Michael Kirley
Peter Harris

**LOCATION:**
Melbourne

**TIME FRAME:**
September 2008
The pure and simple truth is rarely pure
and never simple.
— Oscar Wilde

See everything; overlook a great deal;
correct a little.
— Pope John XXIII

All models are wrong. Some models are useful.
— George E.P. Box

Dedicated to my parents, for so many reasons.
The aim of this thesis is to provide a modelling approach and simulation framework that allows for emergent dynamics in multi-nephron systems to be studied. The ultimate intent of this research is to provide an approach to renal modelling that is capable of predicting whole-kidney function from the dynamics of individual nephrons, and can therefore be of practical use to clinicians. The contributions of this thesis are:

- A modelling approach—*hierarchical dynamical networks*—which combines complex networks and graph automata into a single modelling framework. This approach explicitly captures the structure and interactions in multi-nephron systems, and decouples the structure and behaviour of the model. This approach allows emergent dynamics to be easily explored and analysed.

- The development of a multi-nephron model that produces valid behaviour and renders the simulation of whole-kidney function from the dynamics of individual nephrons computationally tractable. Using this model, the emergent effects of the couplings and interactions between nephrons can be investigated.

- An investigation into the dynamics of multi-nephron systems that focuses on whole-system and hierarchical properties rather than the dynamics of individual nephrons. As part of this investigation, the dynamics of a 72-nephron system are analysed—a system significantly larger than existing multi-nephron models.

- A study of whole-system stability in response to localised impairments in nephron function. This is the first study of the emergent dynamics of impaired nephron function, and serves as an illustration of how the emergent dynamics produced by renal diseases may be predicted and analysed. The impaired multi-nephron systems are shown to exhibit very stable behaviour, which we contend is a feature of both the model and the kidney proper.

- The computational cost of the model is shown to be low enough that the simulation of whole-kidney function is feasible for the first time. It is also demonstrated that simulations can be easily distributed across multiple computers, resulting in a significant gain in performance. An implementation of the model that supports parallel and distributed execution is presented, based on the Join Calculus.

- In order to predict whole-kidney function, a whole-kidney model must be constructed. This thesis proposes two approaches for automatically generating such models.

I conclude that the modelling and analysis techniques presented in this thesis allow for emergent dynamics to be studied in large multi-nephron systems. This work demonstrates that, for the first time,
simulation of whole-kidney function from the dynamics of individual nephrons is tractable. Furthermore, the work provides a basis for predicting emergent effects of localised renal disease. With the continued development of this model, we hope that significant insight will be gained into the onset, progression and treatment of renal diseases.
DECLARATION

This is to certify that

(i) the thesis comprises only my original work towards the PhD except where indicated in the Preface;

(ii) due acknowledgement has been made in the text to all other material used;

(iii) the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Melbourne, September 2008

_________________________
Robert Moss
It is true that we live in a complex world and strive to solve inherently complex problems, which often do require complex mechanisms. However, this should not diminish our desire for elegant solutions, which convince by their clarity and effectiveness.

— Niklaus Wirth, 1984 Turing Award Lecture

ACKNOWLEDGEMENTS

I thank my supervisors Ed Kazmierczak and Mike Kirley, for their boundless enthusiasm and support, their ideas and suggestions, and for being so generous with their time. Making the difficult seem easy and the impossible merely difficult, I learnt everything I know of the world of research under their tutelage.

I owe a massive debt to Peter Harris, for his exceptional generosity and his readiness to share his expertise in a field to which I was a newcomer. My naive questions were always answered quickly and thoroughly, and his guidance was essential in ensuring that my work was kept relevant to the world of renal modelling.

Don Marsh and Randy Thomas have been very supportive of my efforts, and have provided both great assistance and invaluable feedback on a frequent basis. I sincerely thank them, especially for all of their time that was spent patiently correcting my many beginners’ mistakes.

I discussed many of my ideas and problems with Gavin Baker and Tariq Mahmood, receiving countless gems of advice and many helpful suggestions. Paul Fraser was always on hand to commiserate and empathise about the rigours of being a postgraduate student, and to congratulate the hard-won successes.

While conducting my research, I was financially supported by an Australian Postgraduate Award (APA), funded by the Department of Education, Science and Training. I was also awarded the Stawell Scholarship by the School of Engineering to prepare a paper for publication after the submission of this thesis.

Finally, I thank my parents, who have been supportive not just throughout my candidature, but who have also encouraged me from an early age to develop an interest in research. They are, almost certainly, the only people who will read the published thesis in its entirety.
## CONTENTS

I INTRODUCTION 1
1 INTRODUCTION 3
   1.1 Aim 4
   1.2 Overview 5

II BACKGROUND 7
2 THE KIDNEY 9
   2.1 Physiology of the kidney 9
   2.2 Renal modelling 12
   2.3 Disease in the kidney 14
      2.3.1 Symptoms of renal disease 14
      2.3.2 The economic impact of renal disease 16
   2.4 Modelling renal disease 16
   2.5 Modelling the renal physiology as a network 17

3 NETWORK MODELS 19
   3.1 Graph automata models 19
   3.2 Complex network models 21
   3.3 Hierarchy in real-world networks 23
   3.4 Disturbed networks and disease 25

III MODELS AND EXPERIMENTS 27
4 MODEL DESIGN 29
   4.1 Modelling approach 29
      4.1.1 The effects of a discrete time model 30
      4.1.2 Analysis of discrete network automata 32
   4.2 Model structure 33
      4.2.1 The single-nephron model 33
      4.2.2 The multi-nephron model 34
      4.2.3 The multi-column model 36
   4.3 Network properties 38
   4.4 State space equations and update rules 40
      4.4.1 Choosing an appropriate time-step size 40
      4.4.2 Discrete fluid flow and solute transport 41
      4.4.3 Model input 43
      4.4.4 State variables 43
      4.4.5 State update rules 44
   4.5 Summary 53

5 MODEL DYNAMICS 55
   5.1 The single-nephron model 56
   5.2 Two-nephron systems 62
   5.3 An inhomogeneous 8-nephron system 67
   5.4 A 72-nephron multi-column model 73
   5.5 Summary of experiments 74

6 EMERGENT EFFECTS OF LOCALISED IMPAIRMENT 79
   6.1 Incorporating impairments into the model 79
      6.1.1 Stochastic impairments 80
      6.1.2 Stochastic spread of impairments 80
      6.1.3 Deterministic impairments 81
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2</td>
<td>Choice of impairment</td>
<td>81</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Implementation of the impairment</td>
<td>82</td>
</tr>
<tr>
<td>6.3</td>
<td>Measuring the effects of impairments</td>
<td>82</td>
</tr>
<tr>
<td>6.4</td>
<td>Impairments in an 8-nephron system</td>
<td>83</td>
</tr>
<tr>
<td>6.4.1</td>
<td>A single impaired nephron</td>
<td>84</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Two impaired nephrons</td>
<td>88</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Summary of experiments</td>
<td>92</td>
</tr>
<tr>
<td>6.5</td>
<td>Impairments on a larger scale</td>
<td>93</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Impaired nephron pairs</td>
<td>94</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Impaired columns</td>
<td>99</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Summary of experiments</td>
<td>106</td>
</tr>
<tr>
<td>6.6</td>
<td>Conclusions about system stability</td>
<td>107</td>
</tr>
<tr>
<td>7</td>
<td>Large-scale simulation</td>
<td>109</td>
</tr>
<tr>
<td>7.1</td>
<td>Introduction</td>
<td>109</td>
</tr>
<tr>
<td>7.2</td>
<td>Performance of the implementation</td>
<td>110</td>
</tr>
<tr>
<td>7.2.1</td>
<td>The single-threaded algorithm</td>
<td>110</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Performance analysis</td>
<td>111</td>
</tr>
<tr>
<td>7.3</td>
<td>Improving performance with concurrency</td>
<td>114</td>
</tr>
<tr>
<td>7.3.1</td>
<td>The concurrent algorithm</td>
<td>114</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Communication in the concurrent algorithm</td>
<td>117</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Performance of the implementation</td>
<td>119</td>
</tr>
<tr>
<td>7.4</td>
<td>Automatic generation of large models</td>
<td>122</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Stochastic generation of large models</td>
<td>122</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Model growth as a rewriting system</td>
<td>123</td>
</tr>
<tr>
<td>IV</td>
<td>Summary of research</td>
<td>127</td>
</tr>
<tr>
<td>8</td>
<td>Discussion and conclusion</td>
<td>129</td>
</tr>
<tr>
<td>8.1</td>
<td>Contributions</td>
<td>129</td>
</tr>
<tr>
<td>8.2</td>
<td>Limitations and further work</td>
<td>130</td>
</tr>
<tr>
<td>8.2.1</td>
<td>Improved understanding of whole-kidney function</td>
<td>130</td>
</tr>
<tr>
<td>8.2.2</td>
<td>Modelling of renal disease</td>
<td>131</td>
</tr>
<tr>
<td>8.2.3</td>
<td>Pluggable virtual kidney</td>
<td>131</td>
</tr>
<tr>
<td>8.2.4</td>
<td>Modelling kidney failure with reliability factors</td>
<td>132</td>
</tr>
<tr>
<td>8.3</td>
<td>Conclusion</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>133</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Figure 1</td>
<td>Physiology of the kidney.</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2</td>
<td>The functional segments of the nephron.</td>
<td>10</td>
</tr>
<tr>
<td>Figure 3</td>
<td>The inverse-sigmoid signal generated by the TGF mechanism.</td>
<td>11</td>
</tr>
<tr>
<td>Figure 4</td>
<td>An example of a 1D binary cellular automaton.</td>
<td>19</td>
</tr>
<tr>
<td>Figure 5</td>
<td>The beehive rule.</td>
<td>20</td>
</tr>
<tr>
<td>Figure 6</td>
<td>A logic gate implemented with the beehive rule.</td>
<td>21</td>
</tr>
<tr>
<td>Figure 7</td>
<td>The iterative construction of a hierarchical network.</td>
<td>24</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Choosing an appropriate sampling rate.</td>
<td>31</td>
</tr>
<tr>
<td>Figure 9</td>
<td>The single-nephron model.</td>
<td>34</td>
</tr>
<tr>
<td>Figure 10</td>
<td>The single-nephron tubule as a 1D network.</td>
<td>35</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Solute transport in the loop of Henle.</td>
<td>35</td>
</tr>
<tr>
<td>Figure 12</td>
<td>The two-nephron network.</td>
<td>35</td>
</tr>
<tr>
<td>Figure 13</td>
<td>A single column of interstitial fluid.</td>
<td>36</td>
</tr>
<tr>
<td>Figure 14</td>
<td>The arterial tree for a multi-nephron model.</td>
<td>37</td>
</tr>
<tr>
<td>Figure 15</td>
<td>The arterial tree for multiple columns of interstitial fluid.</td>
<td>37</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Diffusion between columns of interstitial fluid.</td>
<td>38</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Evolution of small-world network properties I.</td>
<td>39</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Evolution of small-world network properties II.</td>
<td>40</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Discrete fluid flow along tubule segments.</td>
<td>41</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Fluid flow into a sub-divided tubule segment.</td>
<td>42</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Solute transfer in the loop of Henle.</td>
<td>48</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Oscillations in medullary sodium concentration.</td>
<td>50</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Regular oscillations in single-nephron glomerular filtration rate (sngfr).</td>
<td>57</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Construction and maintenance of a medullary sodium gradient.</td>
<td>58</td>
</tr>
<tr>
<td>Figure 25</td>
<td>The single-nephron dynamics in response to decreased blood sodium concentration (k = 0.3 and PA = 110).</td>
<td>60</td>
</tr>
<tr>
<td>Figure 26</td>
<td>The change in sodium concentration in the top (outer medulla) and bottom (inner medulla) layers of medullary fluid, in response to decreased blood sodium concentration.</td>
<td>60</td>
</tr>
<tr>
<td>Figure 27</td>
<td>The single-nephron dynamics in response to increased blood sodium concentration (k = 0.3 and PA = 110).</td>
<td>61</td>
</tr>
<tr>
<td>Figure 28</td>
<td>The change in sodium concentration in the top (outer medulla) and bottom (inner medulla) layers of medullary fluid, in response to increased blood sodium concentration.</td>
<td>61</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Mean difference in filtration rates in a two-nephron system, over the parameter space (PA, k).</td>
<td>62</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Lyapunov stability over the parameter space (PA, k).</td>
<td>63</td>
</tr>
</tbody>
</table>
Figure 31  The change in SNGFR in response to changes in $P_A$. 64
Figure 32  The effect of varying the vascular coupling between two identical nephrons I. 65
Figure 33  The effect of varying the vascular coupling between two identical nephrons II. 65
Figure 34  The effect of varying the vascular coupling between two inhomogeneous nephrons. 66
Figure 35  The medullary sodium gradient in a two-nephron system when blood sodium concentration remains at 135–145 mmol/L. 68
Figure 36  The medullary sodium gradient in a two-nephron system when blood sodium concentration progressively drops to 110–120 mmol/L. 68
Figure 37  The medullary sodium gradient in a two-nephron system when blood sodium concentration progressively rises to 160–170 mmol/L. 68
Figure 38  Structure of the 8-nephron model. 69
Figure 39  The medullary salt gradient in an eight-nephron system when blood sodium concentration is altered. 70
Figure 40  The salt concentration in the inner and outer medulla of an eight-nephron system when blood sodium concentration is altered. 70
Figure 41  The whole-system filtration rate (WSFR) of an eight-nephron system when blood sodium concentration is altered. 70
Figure 42  The WSFR of eight-nephron systems with different levels of vascular signalling. 71
Figure 43  The WSFR power spectral density of eight-nephron systems with different levels of vascular signalling. 72
Figure 44  The high-level structure of the 72-nephron model. 73
Figure 45  The WSFR of the 72-nephron system when $\gamma = 0$. 75
Figure 46  The individual whole-column filtration rates (WCFRs) in the 72-nephron system when $\gamma = 0$. 75
Figure 47  Power spectral densities of the WSFR of the 72-nephron system for different values of $\gamma$. 76
Figure 48  Power spectral densities of the WCFRs of the 72-nephron system for different values of $\gamma$. 77
Figure 49  Average WCFRs and WSFRs of the 72-nephron system for different values of $\gamma$. 77
Figure 50  The loops of Henle in the eight-nephron system. Each nephron is identified by a number from 1 to 8, and the length of the loops are rated on a scale of 0 to 3. 83
Figure 51  Variations in the WSFR of an eight-nephron system, in response to single impaired nephrons of different loop lengths. 85
Figure 52  Comparisons of WSFR between the original (unimpaired) eight-nephron system and the eight-nephron system where nephron 7 is impaired and $D_P = 0.5$. 86

Figure 53  The power spectral densities of SNGFR in the eight-nephron system with a single impaired nephron. 87

Figure 54  The difference in power spectral densities of SNGFR between the original eight-nephron system and the system with one impaired nephron. 87

Figure 55  Variations in the WSFR of an eight-nephron system, in response to pairs of impaired nephrons of different loop lengths. 89

Figure 56  Comparisons of WSFR between the original eight-nephron system and the eight-nephron system where both longest-looped nephrons are impaired and $D_P = 0.5$. 90

Figure 57  The power spectral densities of SNGFR in the eight-nephron system where both longest-loop nephrons are impaired. 91

Figure 58  The difference in power spectral densities of SNGFR between the original eight-nephron system and the system with two impaired nephrons. 91

Figure 59  The high-level structure of the 72-nephron model. 93

Figure 60  Variations in the WSFR in the 72-nephron system where all nephrons of a specific loop length are impaired. 95

Figure 61  Comparison of the WSFR in the original 72-nephron system to that of the 72-nephron system where all nephrons of a specific loop length are impaired. 96

Figure 62  A comparison of the power spectral densities of WCFR between the original 72-nephron system and the system where all nephrons of a specific loop length are impaired. 97

Figure 63  The difference in power spectral densities of WCFR between the original 72-nephron system and the system where all nephrons of a specific loop length are impaired. 98

Figure 64  Variations in WCFR in the 72-nephron system where all nephrons of loop length 3 are impaired. 99

Figure 65  Variations in the WSFR in the 72-nephron simulations where all of the nephrons in a particular column $(x, y)$ are impaired. 100

Figure 66  Comparison of the WSFR in the original 72-nephron system and the system where all of the nephrons in column 9 are impaired. 101

Figure 67  A comparison of the spectral densities of WCFR between the original 72-nephron system and the system where all of the nephrons in column 9 are impaired. 102
Figure 68  The difference in power spectral densities of \textit{WCFR} between the original 72-nephron system and the system where all of the nephrons in column 9 are impaired. 103

Figure 69  The difference in the power spectral density of column 9 in the original 72-nephron system and the system where all of the nephrons in column 9 are impaired. 104

Figure 70  Variations in \textit{WCFR} in the system where all of the nephrons in column 9 are impaired. 105

Figure 71  Comparing 1-column, 2-column and 4-column simulations. 112

Figure 72  Execution time is $O(n^3)$ for \textit{n} nephrons per column. 113

Figure 73  The master/slave relationship between columns. 115

Figure 74  Communication between the controller and columns. 116

Figure 75  Communication between columns when performing the inter-column diffusion. 117

Figure 76  Reordering the communication in Fig. 74, in order to reduce the delay between time-steps. 117

Figure 77  Amdahl’s law. 121

Figure 78  The improvement in performance as the simulation is distributed over multiple computers. 121

Figure 79  A Lindenmayer system that resembles the arterial tree of the kidney. 125

Figure 80  A simple model of compliant vessels. 131

LIST OF TABLES

Table 1  Some of the solutes regulated by the nephron. 10
Table 2  Properties of the \textit{n}-nephron single-column network. 39
Table 3  Properties of the \textit{n}-nephron $c^2$-column network. 40
Table 4  The solutes support by the model. 43
Table 5  State variables in the nephron model. 45
Table 6  Model parameters. 45
Table 7  The behaviour of each tubule segment in the nephron model. 46
Table 8  Notation for model equations. 47
Table 9  Distal tubule responses to hormone concentrations in the filtrate. 52
Table 10  Steady-state concentration levels. 56
Table 11  Systematic decrease blood sodium concentration. 60
Table 12  Systematic increase in blood sodium concentration. 61
Table 13  The difference in the sodium gradient at standard and altered blood sodium concentrations. 70
Table 14 Increases in spectral density peaks in the SNGFR of nephrons in the system with one impaired nephron. 86
Table 15 Increases in spectral density peaks in the SNGFR of nephrons in the system with two impaired nephrons. 92
Table 16 Performance comparison of two models of identical size. 114
Table 17 Message content in inter-process communication. 118

ACRONYMS

AAR afferent arteriole resistance
ADH antidiuretic hormone
ARF acute renal failure
BOINC Berkeley Open Infrastructure for Network Computing
CKD chronic kidney disease
CPU central processing unit
CRF chronic renal failure
DCT distal convoluted tubule
ECF extracellular fluid
EPO erythropoietin
ESRF end-stage renal failure
GFR glomerular filtration rate
GN glomerular nephritis
IgA Immunoglobulin A
RAS renin-angiotensin system
RRT renal replacement therapy
SIS susceptible-infected-susceptible
SNGFR single-nephron glomerular filtration rate
TGF tubuloglomerular feedback
THPO thrombopoietin
WCFR whole-column filtration rate
WSFR whole-system filtration rate
Part I

INTRODUCTION
INTRODUCTION

A complex system is a system comprised of many interconnected parts, whose behaviours are both variable and dependent on the behaviour of the other parts [161, 146]. Such systems exhibit emergent behaviour—behaviour that may be deterministic, but which has properties that can only be examined at a higher level than the individual parts [3]. The difficulty in understanding such systems is that the behaviour of the whole is highly dependent on the individual parts, and each part is highly dependent on other parts [17]. More simply, a complex system is holistic—it is more than just the sum of its parts—and we must understand how the interactions between the parts affect the whole.

The kidney is one of the major organs responsible for homeostasis of the body. It regulates acid-base balance, extracellular fluid, electrolyte concentrations, blood volume and blood pressure, as well as excreting metabolic waste products. The kidney is also a complex system [133, 173, 164], consisting of approximately 800,000 to 1,000,000 nephrons in the human [149]. The nephron is the basic filtration unit of the kidney, and is a convoluted tubule that adjusts the solute levels of the body’s blood plasma. Nephrons are surrounded by interstitial fluid, into which solutes are secreted and reabsorbed, and by a complicated network of capillaries that also exchange solutes with the surrounding interstitial fluid. While the behaviour of individual nephrons can fluctuate widely and can even behave in a chaotic manner [112], the overall behaviour of the kidney remains stable. In particular, interactions occur between neighbouring nephrons that give rise to synchronisations in pressure and flow oscillations—indicating that groups of nephrons are capable of producing emergent behaviour [166, 81, 112, 145, 167].

In fact, the kidney is able to maintain its homeostatic functions over an astonishing range of bodily conditions, such as wide variations in blood pressure, extracellular fluid volume and solute concentrations. The kidney is also resilient to the effects of renal diseases and the loss of renal tissue (e.g. surgical removal)—a human can survive with only one half of a single kidney. Although some of the regulatory mechanisms in the kidney are well understood, much less is known about how the functional stability of the entire kidney arises from the individual components and their interactions.

To the best of our knowledge, there has been relatively little research into how the many regulatory mechanisms affect higher-level behaviour in the kidney. Researchers have focused on understanding the chemical processes and fluid mechanics at the cellular level, and modelling the kidney’s vascular system. Many low-level properties of nephron tubule segments have been modelled [189, 177] and a number of multi-nephron models have been proposed (mostly two-nephron models [78, 81, 145], although a 22-nephron model also exists [112]). However, the standard methods and techniques in this field are not amenable to modelling and analysing the kidney as a
complex system [17, 201], due to the coupling between equations, the complexity of calculating solutions, and the inability to decouple the structure and behaviour of the model. This thesis proposes a different approach that minimises the coupling between equations, reduces the computational cost of solving the model equations, and decouples the model’s structure from its behaviour.

1.1 AIM

A knowledge gap exists between the understanding of the cellular processes and transporters that give rise to nephron function, and the understanding of whole-kidney function. There has been little research into how the dynamics of individual nephrons—and the couplings, interactions and regulator mechanisms that arise between nephrons—give rise to the dynamics observed at the whole-kidney level.

The aim of this thesis is to address this gap by modelling the kidney as a complex system and analysing it accordingly. The ultimate intent of this research is to provide an approach to renal modelling that is capable of predicting whole-kidney function from the dynamics of individual nephrons, and can therefore be of practical use to clinicians. Specifically, this thesis makes the following contributions:

• A modelling approach—*hierarchical dynamical networks*—which combines complex networks and graph automata into a single model. This approach explicitly captures the structure and interactions in multi-nephron systems, and decouples the structure and behaviour of the model. This approach allows emergent dynamics to be easily explored and analysed.

• The development of a multi-nephron model that produces valid behaviour and renders the simulation of whole-kidney function computationally tractable. Using this model, the emergent effects of the couplings and interactions between nephrons can be investigated.

• An investigation into the dynamics of multi-nephron systems that focuses on whole-system and hierarchical properties rather than the dynamics of individual nephrons. As part of this investigation, the dynamics of a 72-nephron system are analysed—a system significantly larger than existing multi-nephron models.

• A study of whole-system stability in response to localised impairments in nephron function. This is the first study of the emergent dynamics of impaired nephron function, and serves as an illustration of how the emergent dynamics produced by renal diseases may be predicted and analysed. The impaired multi-nephron systems are shown to exhibit very stable behaviour, which we contend is a feature of both the model and the kidney proper.

• The computational cost of the model is shown to be low enough that the simulation of whole-kidney function is feasible for the first time. It is also demonstrated that simulations can be easily
distributed across multiple computers, resulting in a significant gain in performance. An implementation of the model that supports parallel and distributed execution is presented, based on the Join Calculus [59, 60].

• In order to predict whole-kidney function, a whole-kidney model must be constructed. This thesis proposes two approaches for automatically generating such models.

1.2 Overview

Renal physiology is introduced in Chapter 2, followed by a survey of existing tubule, whole-nephron and multi-nephron models. These models, which are able to capture precise low-level dynamics of the nephron tubule, are shown to be unsuitable for modelling and analysing the kidney as a complex system. This chapter also presents a brief discussion of renal disease and a review of renal disease studies.

Graph automata and complex networks are introduced in Chapter 3, and are shown to be better suited to modelling the kidney as a complex system than the renal models presented in Chapter 2. Existing graph automata and complex network models are reviewed, followed by a discussion of how hierarchical properties of complex systems can be captured with such models. The chapter concludes by reviewing the application of these models to studying the effects of disease.

The modelling approach—**hierarchical dynamical networks**—is presented in Chapter 4 and contrasted to continuous non-hierarchical models. The structure of the multi-nephron model is presented and compared to the structure of existing networks, and the update rules of the model are then given.

Chapter 5 presents the experimental design used to validate the model and to explore the emergent dynamics produced by the model. This is followed by the results of each experiment in turn: the validation of the single-nephron dynamics; the validation of the two-nephron dynamics; a further exploration of the interactions between competing coupling mechanisms in the two-nephron system; and finally an analysis of the interactions between competing coupling mechanisms in 8-nephron and 72-nephron systems.

A study of whole-system stability in response to localised impairments in nephron function is presented in Chapter 6. The chapter begins by discussing how localised impairments can be incorporated into the model, and how the emergent effects of such impairments can be studied. A simple impairment is introduced into 8-nephron and 72-nephron systems and the stability of these systems is analysed. The impaired multi-nephron systems are shown to exhibit stable behaviour, which we contend is a feature of both the model and the kidney proper.

The implementation of the model simulation is presented in Chapter 7 and the performance of this implementation is analysed, demonstrating that the simulation of whole-kidney function from the dynamics of individual nephrons is computationally tractable. It is also demonstrated that the simulations can easily be distributed across
multiple computers—a distributed implementation based on the Join Calculus [59, 60] is presented and a performance analysis reveals that the result in a significant gain in simulation performance. A whole-kidney model is required to predict whole-kidney function, and the chapter concludes by proposing two methods for automatically generating such models.

The contributions of this thesis are summarised in Chapter 8. The limitations of this work are then discussed, and potential avenues of further research to build on the work are proposed. Finally, the research outcomes are evaluated against the research aim.
Part II

BACKGROUND
THE KIDNEY

This chapter begins by introducing the reader to the physiology of the kidney—the structure and function of the kidney is briefly presented—and a review of existing nephron tubule models is presented. This is followed by an overview of the major renal diseases, their significance, and their effects on the kidney, and a review of models of kidney disease. The chapter is concluded with a short description of the modelling approach that will be used to develop the nephron model presented in this thesis.

2.1 PHYSIOLOGY OF THE KIDNEY

The kidney is one of the major organs involved in whole-body homeostasis. It is responsible for regulating solute concentrations in the blood, the volume, composition and avid-base balance of extra-cellular fluid, and blood pressure in the body. The kidney also plays a part in the body’s endocrine system, secreting hormones such as renin as part of the renin-angiotensin system (RAS), erythropoietin (EPO) to regulate red blood cell production, and thrombopoietin (THPO) to regulate platelet production.

The functional unit of the kidney is the nephron, a long, segmented tubule into which blood plasma is filtered (see Fig. 1a). As the filtrate flows along the nephron tubule, water is reabsorbed into the bloodstream and solutes are both reabsorbed into the bloodstream and secreted into the tube (see Table 1). The filtrate flows from the nephron into the collecting ducts (labelled “A” in Fig. 1a), which drain the fluid into the bladder via the urethra. The human kidney contains around one million nephrons [149], which are arranged in renal lobes (see Fig. 1b).

(a) A single nephron and the peritubular capillaries. (b) Nephrons (shown as glomeruli connected to arcuate arteries) are grouped in renal lobes.

Figure 1: Physiology of the kidney.
<table>
<thead>
<tr>
<th>Solute</th>
<th>Formula</th>
<th>Reabsorbed</th>
<th>Secreted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sodium</td>
<td>Na$^+$</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Potassium</td>
<td>K$^+$</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Proton</td>
<td>H$^+$</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Chloride</td>
<td>Cl$^-$</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Calcium</td>
<td>Ca$^{2+}$</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Bicarbonate</td>
<td>HCO$_3^-$</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Phosphate</td>
<td>PO$_4^{3-}$</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Urea</td>
<td>(NH$_2$)$_2$CO</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Glucose</td>
<td>C$<em>6$H$</em>{12}$O$_6$</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Amino acids</td>
<td>various</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1: Some of the solutes regulated by the nephron.

![Diagram showing the functional segments of the nephron.](image)

Figure 2: A diagram showing the functional segments of the nephron. 1: the afferent arteriole, which provides the nephron with blood; 2: the glomerulus, a network of capillaries that filters a fraction of the blood plasma into the nephron tubule (the remaining blood flows into the peritubular capillaries that surround the tubule—not shown here, but shown in Fig. 1a); 3: the proximal tubule; 4: the descending limb of Henle; 5: the thin ascending limb of Henle; 6: the thick ascending limb of Henle; 7: the distal tubule; 8: the collecting ducts, which drains the filtrate into the bladder; 9: the TGF control mechanism (an interaction between the macula densa and the filtration process).

The structure of a single nephron is shown in Fig. 2. The nephron is surrounded by connective tissue, which holds the nephron tubules and renal blood vessels, and the interstitial fluid, which can be divided into compartments within the cortex, the outer medulla and the inner medulla. Blood flows through the afferent arteriole (1) and into the glomerulus (2), where a fraction of the blood plasma is filtered into the nephron tubule, and the remaining blood flows into the peritubular capillaries (not shown). The filtrate—the plasma that has been filtered into the tubule—flows through the proximal tubule (3), the descending and ascending limbs of the loop of Henle (4–6),...
the distal tubule (7) and finally the collecting ducts (8). The loop of Henle acts as a counter-current exchange mechanism for NaCl, increasing the NaCl concentration in the interstitial fluid as the loop descends. In addition to this counter-current mechanism, the production of lactate in the tubule may contribute significantly to the accumulation of sodium [76], and solutes such as urea also play a role in the concentrating mechanism [97, 98, 176]. Existing models have incorporated gradients—measured between the top and bottom of the loop of Henle—such as 150mmol/L [78], although the gradient can be as large as 1200mmol/L in human kidneys [169] and over 2000mmol/L in other species [158, 76].

A major regulation mechanism (9) in the kidney is tubuloglomerular feedback (TGF), a negative-feedback mechanism that regulates the filtration rate of an individual nephron. A signal is sent to the afferent arteriole that controls its diameter (and thus the filtration rate), based on the delivery of solutes, mainly sodium chloride, to a region called the macula densa (the end of the ascending limb of Henle, 6–7 in Fig. 2). The filtration rate affects distal sodium delivery—high sodium levels indicate that the filtration rate is too high for sodium to be reabsorbed effectively, while low sodium levels indicate that the filtration rate is so low that too much sodium is being reabsorbed—and the TGF mechanism keeps the filtration rate within appropriate bounds (see Fig. 3). There is evidence that the TGF mechanism may also be mediated through the resistance of the efferent arteriole [151, 68, 23]—the arteriole that carries blood out of the glomerulus—and the permeability of the glomerular capillaries [130, 28, 181].

Figure 3: The inverse-sigmoid signal generated by the TGF mechanism. Changes in the solute delivery to the macula densa result in an inverse response in the filtration rate. The system typically oscillates in the highlighted region, about the equilibrium point indicated on the graph.

Specialised cells in the tubule wall generate a signal that controls the diameter of the afferent arteriole (which supplies blood to the glomerulus). The signal is not instantaneous, but has a delay of around
3–5 seconds [166, 145, 79]), and there is a further delay for the filtrate to flow from the glomerulus to the macula densa (around 10–15 seconds [166, 78, 79]). This negative-feedback mechanism has been the focus of many multi-nephron models, and is known to have an inverse-sigmoid relationship [160, 166] with sodium delivery [159] at the macula densa (see Fig. 3).

Two different types of interactions are known to occur between nephrons in multi-nephron systems [166, 81, 112]: vascular signaling, where the TGF signal from one nephron also reaches nearby nephrons; and hemodynamic coupling, where the resistance of an afferent arteriole affects the hydrostatic pressure at nearby nephrons. Vascular signaling favours in-phase synchronisation [112], since an increase or decrease in the TGF signal of one nephron is propagated to neighbouring nephrons. Conversely, hemodynamic coupling favours anti-phase synchronisation [112], since the contraction of an afferent arteriole reduces the filtration rate of that nephron and increases the flow to—and therefore the filtration rate of—the neighbouring nephrons. One model has predicted that vascular signalling overwhelms the hemodynamic coupling, leading to in-phase synchronisations in adjacent nephrons [112].

The kidney has a hierarchical physical composition—the nephron consists of a number of distinct segments; nephrons can be grouped into interacting neighbourhoods (such as all nephrons joined to a single cortical radial artery [112]); large numbers of these neighbourhoods form renal pyramids; and the kidney consists of multiple such pyramids (e.g., from 8–18 in humans [55] to a single pyramid in smaller animals such as rodents and monotremes [211]). The functional structure of the kidney is closely related to its physical structure—witness how the arrangement of the nephron tubule reflects the functional structure of the TGF mechanism—and therefore exhibits a similar hierarchy. Hierarchy is a property of many complex systems [164, 150, 47] and it would be desirable to capture this hierarchy explicitly in our model, as opposed to existing nephron models, which only capture hierarchy implicitly (if at all). Hierarchy in models is discussed further in Sect. 3.3.

2.2 Renal Modelling

The study of renal physiology takes three forms: clinical studies that focus on diagnosing and treating kidney-related diseases; experimental studies that examine the behaviour and function of the kidney (in vivo or in vitro); and theoretical studies that propose models to capture the behaviours and functions that have been observed experimentally. These fields encompass the study of more than just the behaviour of the nephron tubule segments, and can be divided into three general areas: cellular, tubular and vascular physiology.

For the purpose of this thesis it is sufficient to concentrate on tubular physiology, since the models presented herein focus on the behaviour of and interactions between nephrons. To this end, the range of theoretical models that have been proposed are now surveyed, as they not only provide a starting point for the design of new models, but also compare the models to the relevant portion
of the available experimental data. Several clinical studies are also reviewed in Sect. 2.4.

The majority of nephron modelling has focused on replicating low-level behaviour in individual tubule segments (e.g., the proximal tubule [190, 193], distal convoluted tubule [192, 192, 38, 37] and collecting ducts [191, 191]). Experimental data shows that in the nephron, the flow rate and tubule pressure both oscillate, and models have shown that such oscillations can arise due to the TGF mechanism [78, 80]. Existing models are able to reproduce a wide range of other experimental data, but uncertainty still remains about the roles of transporters along the nephron tubule, due to the limitations of existing experimental techniques and the inability of current models to capture the essence of all observed phenomena [189].

There are some published models that study the behaviour of multiple nephrons, such as their ability to regulate concentration gradients in the medullary fluid [76, 108], or synchronisation phenomena that can arise due to pressure oscillations in blood flow [81, 112]. These models do not include the entire nephron, although at least one whole-nephron model exists [170].

These tubule models are generally formulated as systems of differential equations that model the mechanics of fluid flow along tubule segments and the transport of solutes and water across the tubule walls. This approach has yielded models that are able to replicate specific, detailed behaviours.

For example, Weinstein [192] models the rat early distal convoluted tubule (DCT) as a tubule lined by DCT cells and intercellular space (both of which admit solute transport). Solute transport is governed by electrochemical potential gradients (via cotransporters) or coupled to metabolic energy (via an ATPase). This model updates previous work, in drawing on new experimental data to refine the formulation (and inclusion) of several cotransporters, as well as including a more realistic model of the tubule lining than previous models, allowing cell volume to vary and not requiring the intercellular and peritubular fluids to be at equilibrium. This model provides new observations regarding cell volume homeostasis and new predictions concerning the effects of alkalosis on the DCT.

In short, this paper presents a model of the rat early DCT that is a significant extension of existing DCT models, by virtue of having fewer underlying assumptions and more detailed cotransporter formulations. Despite these features, this model is a development of the published tubule models, and does not break new ground with respect to the approach taken.

Unlike discrete equations, which explicitly calculate the current state of a system based on the previous state of the system, systems of continuous equations only place constraints upon the system. Solving such constraints can be computationally very expensive (often requiring numerical solving methods which discretise the equations). A unique solution may not exist, in which case linear combination of the solutions must be considered.

The accuracy of these kinds of models is not without drawbacks, as the models can be computationally very expensive. Due to the coupled nature of the equations, the addition of another variable or a change
to the equation governing an existing variable can have significant effects on the solutions of other equations in the model. Another drawback is that the tubule models cannot necessarily be linked together to form models of an entire nephron tubule, as they may have different underlying assumptions or incompatible formulations (although some tubule models have been successfully combined). Our approach overcomes these drawbacks by using difference equations, which decouple the continuous equations and provide explicit solutions for each variable.

### 2.3 Disease in the Kidney

Renal diseases can prevent the kidney from functioning correctly, resulting in renal failure. Renal failure is divided into acute and chronic forms. Acute renal failure (ARF) is a rapid loss of renal function due to kidney damage, which may be reversible if it is treated promptly. Chronic kidney disease (CKD) is a progressive loss of renal function over a period of months and years, which is divided into five stages, based on the decrease in the kidney filtration rate. The final stage is referred to as end-stage renal failure (ESRF), and requires some form of renal replacement therapy (RRT)—either a kidney transplant or dialysis. ARF may also result in ESRF. Regardless of the cause, ESRF manifests as an abnormally low glomerular filtration rate (GFR) such that the kidney is unable to perform its function.

This review focuses on diseases that cause CKD. The most common causes of CKD and ESRF in western countries are:

- Diabetic nephropathy—thickening of the glomeruli.
- Hypertension—chronically elevated blood pressure.
- Glomerulonephritis—inflammation of the glomeruli.

Together, these three conditions account for around 75% of all adult cases of chronic renal failure (CRF) [41]. In particular, diabetic nephropathy is the most common cause of ESRF in the USA [178, 53].

#### 2.3.1 Symptoms of renal disease

With the aim of studying how the onset of renal disease in individual nephrons can affect the whole-system dynamics of multi-nephron systems, it is the symptoms of the renal diseases that are of greatest interest, as the symptoms indicate the changes in dynamics caused by the diseases. The most common symptoms of renal disease are now introduced.

**Proteinuria**

Proteinuria is the presence of abnormally large amounts of protein in the urine—which can cause the urine to become foamy—and is a sign of potential renal damage. It was understood that a charge barrier in the glomerulus prevents protein from entering the nephron tubule [180, 185, 39, 135] and that the presence of protein in the urine is due to deterioration in this charge barrier, allowing proteins
such as albumin to enter the nephron tubule and ultimately the urine [69, 25]. However, the influence of this electrostatic interaction has been questioned [45, 138, 73, 155] and there is evidence that protein enters the nephron tubule under normal conditions, and that the presence of protein in the urine is due to the failure of a reabsorption mechanism in the proximal tubule cells [116, 32, 21, 44]. Proteinuria was first described by William Charles Wells [63, 33], who suggested that the serum in the urine was derived from the blood [194] and gave the first definition of the limits of normal albuminuria [195] (similar figures are still used today [33]).

Hematuria

Hematuria is the presence of red blood cells in the urine. Aside from renal diseases, hematuria can indicate the presence of kidney stones or a tumour. Hematuria is a symptom of glomerulonephritic diseases such as Immunoglobulin A (IgA) disease (also known as Berger’s nephropathy), which is the most common type of glomerulonephritis in adults world-wide [65, 20, 128].

Hypoalbuminemia

Hypoalbuminemia is a condition where the level of albumin in the blood is abnormally low. It can be caused as a result of proteinuria, and may also indicate liver failure, cirrhosis or chronic hepatitis. As albumin is the major protein in the human body (contributing around 60% of the total plasma protein by mass), a significant reduction in albumin levels significantly lowers the oncotic pressure of the blood. Such a reduction leads to excessive amounts of water diffusing from the blood into the extra-cellular fluid, causing general swelling of the body (edema).

Hyperlipidemia

Hyperlipidemia is the presence of abnormally large amounts of lipids in the blood, which can arise when the liver responds to albumin loss by secreting more protein into the bloodstream.

Hypertension

Hypertension is a condition where the blood pressure is chronically elevated, and is one of the risk factors for cardiovascular diseases, as well as a major cause of CRF. One of the major causes of hypertension is a diet that is high in sodium [156, 92].

Nephritic Syndrome

Nephritic syndrome is a collection of signs associated with glomerular disorders, characterised by proteinuria, hematuria and hypertension. It is not a specific diagnosis and so the underlying causes can vary, but the common feature is an inflammation of the glomeruli, leading to salt and water retention and ultimately a reduction in kidney function.
Nephrotic Syndrome

Nephrotic syndrome is a disorder where large amounts of protein leak from the blood into the urine. It is characterised by proteinuria, hypoalbuminemia, hyperlipidemia and edema (and occasionally hypertension). Nephrotic syndrome has many specific causes (such as diabetes and glomerulonephritis), but the common mechanism is inflammation causing damage to the glomeruli, allowing proteins such as albumin to pass through the kidneys and into the urine.

2.3.2 The economic impact of renal disease

Incidence of renal disease is becoming more prevalent in both western and developing nations, placing a rapidly growing burden on global health care [36, 13, 183, 184]. Long-term kidney disease also increases the risk of developing further complications such as cardiovascular diseases, respiratory infections, bone and muscle problems, and anaemia [13]. Once ESRF occurs, the only option is to pursue palliative treatments known as RRT. Dialysis is the most common form of treatment—approximately 77% of global RRT patients undergo dialysis, with the remaining 23% having functional transplants [125]—but it is also the most costly form of treatment [62] and is not without drawbacks [154].

The burden of premature death from diabetes is similar to that of HIV/AIDS, with one in 20 deaths attributable to diabetes, and the number of people with diabetes is expected to more than double by 2030 [136].

These predictions of increased incidence of renal disease and the associated increase in health-care costs highlight the importance of improving our understanding of how renal diseases affect renal function. Studying the effects of renal diseases through the use of models may lead to improvements in the diagnosis and treatment of renal diseases, helping to reduce the financial and social impacts of these diseases.

2.4 Modelling renal disease

The causes and mechanisms of renal disease have been investigated at the cellular level [21, 32] and many clinical experiments [10, 86, 114, 58, 91], laboratory experiments [95, 24, 84, 144] and case studies [172, 14, 148] have been performed. Rashidi and Khodarahmi present a simple model of nephron loss in response to CRF [149], but there are few other models that attempt to capture the effects of renal disease on the kidney or on multi-nephron systems. In particular, it does not appear that any existing tubule or nephron models have attempted to model the impact of renal disease on the nephron in the context of multi-nephron systems, precluding the study of emergent dynamics produced in the presence of, or in response to, renal disease.

Thomson et al. present “the tubular hypothesis of glomerular filtration” [178], which uses a control theory model to explain several nuances of kidney function in early diabetes. Early diabetes is known to have strange effects on GFR, including diabetic hyperfiltration, a
paradoxical effect of dietary sodium, and the renal response to dietary protein and amino acid infusion. The tubular hypothesis presumes that these effects stem from primary effects on the proximal tubule and loop of Henle, which impact the single-nephron glomerular filtration rate (SNGFR) by feedback through the macula densa and the TGF mechanism. Such primary effects are categorised as "tubular" in comparison to all other possible effectors, which are classified as "vascular". The difference between tubular and vascular effectors is that vascular effectors have the same effect on GFR and sodium delivery to the macula densa—if the GFR is increased, so is the sodium delivery—while tubular effectors have contrary effects—if the GFR is increased, the sodium delivery is decreased.

The published data on the content of the early distal tubule in diabetes consistently lists sodium values substantially below normal, and experimental data shows that diabetic nephrons respond to this with hyperfiltration [72]. However, there is also a change in the TGF response which indicates that a vascular effect may have almost as much control over GFR.

The diabetic kidney also exhibits a "salt paradox" where GFR varies inversely with dietary sodium intake, resulting in a lower GFR when sodium intake is increased, rather than the normal response of an elevated GFR. This cannot be explained by any vascular effectors, but can be explained by the tubular hypothesis as long as the proximal tubule is the dominant mechanism linking GFR to extracellular fluid (ECF).

Glycine infusion normally causes GFR to increase without altering sodium levels at the macula densa, indicating that both vascular and tubular effects contribute to the increase in GFR, while in the diabetic kidney the vascular response is minimal. Protein feeding also increases GFR, by increasing tubular reabsorption to levels that decrease the sodium delivery to the macula densa, despite the elevated GFR. This tubular effect remains present in the diabetic kidney.

In summary, Thomson et al. demonstrate that this tubular hypothesis and the associated control theory model can explain several factors that influence the behaviour of the kidney in early diabetes. Their paper does not focus on specific cellular mechanisms, preferring to treat the kidney as an integrated system of parts—an approach that is reminiscent of our own. The major difference is that this model starts from experimental observations of the whole kidney to suggest the effects that early diabetes has on the nephron tubule, while the focus of our disease modelling is to start from known effects in the nephron tubule and to use our models to determine the resulting behaviour at a higher level, such as the whole kidney.

2.5 MODELLING THE RENAL PHYSIOLOGY AS A NETWORK

As will be discussed in the following chapter (Chapter 3), network models are well suited to capturing both implicit and explicit hierarchies present in the system being modelled, and are also well suited to studying the changes in system dynamics that arise in response to alterations in the model structure and low-level behaviours. In particular, network models provide an advantage when exploring the
effects of different couplings and interactions between parts of the model, as these mechanisms are explicitly captured by the model structure.

Furthermore, due to the use of discrete time equations, such models are able to scale to large systems without incurring the increase in computational cost that is typical of continuous systems. This point is discussed further in Sect. 4.1.1 and the model presented in this thesis is demonstrated to scale to extremely large models in Chapter 7.
As mentioned in the introduction, there are two common classes of network model—networks with regular, fixed topologies and nodes whose states evolve due to homogeneous update rules [201]; and networks with irregular topologies, where nodes have constant state and dynamics are realised as changes in the network topology [139]. However, the kidney has both an irregular static functional topology (closely tied to its physical structure) and localised states that evolve inhomogeneously over time. The model presented in this thesis will therefore have features in common with both networks, and so both classes of network models are examined in this chapter. Firstly, definitions of cellular and graph automata are presented, followed by a review of existing automata models. This is followed by an introduction to common network properties and a review of existing complex networks models.

### 3.1 Graph Automata Models

A cellular automata $A = \{S, C, N, f\}$ consists of a finite state space $S$, a configuration of the cells $C \in S^Z$ for a finite of infinite discrete regular grid of cells $Z$, a neighbourhood function $N : Z \rightarrow S^n$ and an update rule $f : S^n \rightarrow S$. Time is discrete and at any time $t$, the configuration of an automaton is $C(t) \in S^Z$. For each cell $c \in Z$, the state at time $t+1$ is calculated from the state of the neighbourhood $N(c)$ at time $t$, as shown in Eq. 3.1.

$$c(t + 1) = f( N(c) )$$

(a) An update rule—cells change state when their neighbours' states differ. 
(b) The update rule applied to an automaton.

Figure 4: An example of a 1D binary cellular automaton that has a ring topology—the cells at both ends are connected together.

An example of a 1D automaton with a binary state space is shown in Fig. 4. The update rule $f$ for this automaton (Fig. 4a) changes the state of a cell when the states of the two neighbouring cells differ. An application of this rule to a configuration is shown in Fig. 4b.

Graph (or network) automata are an extension of cellular automata, where the topology of the cells $Z$ is generalised to a network, rather than being restricted to a regular grid.
A graph (or network) $G = (V, E)$ consists of a set of nodes ($V$), and edges that connect pairs of nodes ($E \subseteq V \times V$). A graph is by default undirected, meaning that the edges $(v_1, v_2)$ and $(v_2, v_1)$ are equivalent, and for any $(v_1, v_2) \in E$ it is also true that $(v_2, v_1) \in E$. Graphs may also be directed, in which case the edges $(v_1, v_2)$ and $(v_2, v_1)$ are distinct, and an edge $(v_1, v_2)$ is said to be directed from $v_1$ to $v_2$. The neighbourhood of a cell (node) $c$ in a graph automata is the set of cells (nodes) that are directly connected to $c$ in the graph.

Graph automata have been applied to the study of biological phenomena, such as computational operations in reaction-diffusion systems [2] and genetic regulatory networks [203].

Adamatzky et al. [2] show how basic computational operations can be implemented in a three-state hexagonal automaton. This automaton models a reaction-diffusion system using the “beehive rule” [206], which is presented in Fig. 5. The beehive rule exhibits glider dynamics—that is, starting from a random configuration, it produces regular patterns that move across the cells whilst retaining their shape (i.e., gliders). Adamatzky et al. show that by interpreting the direction in which the gliders travel as truth values, the beehive rule is capable of producing logic gates (as depicted in Fig. 6), and performing actions such as multiplication. The result is that the automaton is logically universal—any sequential machine can be constructed in the system.

![Figure 5](beehive_rule.png)

(a) The beehive update rule in matrix form

(b) An example of the beehive rule, for a cell that has two S1 neighbours and two S2 neighbours.

Figure 5: The beehive rule operates on a hexagonal grid whose cells have a state-space $S = \{S0, S1, S2\}$ [2].

While this is a significant achievement—it could potentially be a theoretical prototype of physically-realised reaction-diffusion processors—the analysis and interpretation of the automaton’s behaviour is too deeply rooted in visual observations of patterns in the cells. While
such an analysis might be automated once the behaviour and their interpretations are known, it can not be readily applied to finding and analysing patterns of behaviour in larger, more complex models. Indeed, the state-space of a single nephron is so complicated that it is not obvious how to visualise more than a few rudimentary properties (such as changes in concentration of a single solute), which precludes the possibility of discerning behavioural patterns visually.

This failure of analysis techniques to scale to large models is not isolated to this single paper, rather it is typical of the graph automata models. Indeed, Wolfram acknowledges both the necessity of visual analysis and the difficulties it entails [201, 3§12 pp. 111–3], remarking that as a result he ignored the possibility of cellular automata producing complex behaviour for several years.

One other shortcoming of standard graph automata is that all nodes are homogeneous, sharing the same state-space and update rules. As discussed in Sect. 2.1, nephrons and nephron tubule segments are not homogeneous, and so a graph automata model of a nephron must allow for nodes with different state-spaces and update rules.

Although nephrons are inhomogeneous, they do share a common basic structure and set of transport mechanisms. Each tubule segment is typically lined by the same types of cells regardless of which nephron it belongs to—although some tubules may not contain all possible cell types—and so all instances of each segment that share the same cell types can be modelled homogeneously. By taking this approach, one does not obtain a graph automata with homogeneous nodes and update rules, but rather an automata with classes of nodes and update rules.

However, when attention is turned to modelling large groups of nephrons, it becomes clear that the analysis will necessarily be largely statistical in nature, and that the models must have more complex, less regular topologies than the typical graph automata. Such networks are known as Complex Networks.

3.2 Complex Network Models

Large networks with complex topologies have been studied in the context of many biological phenomena, such as neural networks...
[54], protein interactions [90], metabolic reactions [4], and genetic regulation [132]. Some basic network properties are:

**Degree** The number of edges that connect to a node. For directed graphs this can be extended to discern between edges that link to a node (indegree) and edges that link from a node (outdegree).

**Degree Distribution** The relationship between the degree $k$ and the probability $P(k)$ of a given node having that degree.

**Diameter** The maximum length of the shortest paths between each pair of nodes.

**Connection Probability** The probability that two nodes selected at random are connected by an edge.

**Clustering Coefficient** For a specific node $n$, the fraction of the possible edges between all neighbours of $n$ that are present; for a network $N$, the average of the clustering coefficients for each node in the network.

Based on these properties, complex networks can be classified into a number of categories:

**Regular** Each node has the same degree $k$ and the network is said to be $K$-regular.

**Random** Each pair of nodes is connected by an edge with uniform probability $p$. The degree distribution of such networks is approximately Gaussian.

**Small World** The shortest path between two nodes is likely to be small as the shortest path length scales logarithmically or slower with network size $n$ [188]. In contrast, the shortest path length scales linearly with $n$ in a regular network.

**Scale-Free** The network has a power-law degree distribution. Many, perhaps even most, of the real-world networks that have been studied are scale-free [124]. The mechanism proposed as the explanation for this observation is “growth by preferential attachment” [18]—when a node is added to the network, it is connected to an existing node $i$ with a probability proportional to the degree of $i$.

Wang and Chen [186], Newman [132] and Albert and Barabási [4] all review the progress and significant results in the current studies of various complex networks, including the World Wide Web, human language, and paper authorship. The dynamics of these models are derived from their topology, either by adding or removing nodes from the network, or by percolating node states across the network (e.g., to model the spread of disease). One of the main areas of research using these models is the analysis of network robustness\(^1\) in response to changes in the network (e.g., the removal of vertices or a change in propagation parameters).

---

\(^1\) Also called resilience or stability.
The analysis methods used to determine network resilience typically focus on how statistical properties such as reachability are affected by changes to the network. For example, percolation theory is concerned with the propagation of state across a network (e.g., states such as “working” and “failed”). The parameters governing the state propagation are altered and changes in vertex patterns are then analysed for properties such as reachability and connected sub-sets. This approach is widely used for modelling phenomena such as the spread of diseases in societies and the spread of computer viruses over computer networks.

While this approach might initially seem well-suited to our aim of measuring nephron resilience to the onset of renal disease, this is not the case. Renal diseases affect factors such as filtration rates and the strength of transport mechanisms, and it is not at all clear how these effects can be understood or analysed in terms of the usual kinds of statistical network properties.

Also, the manner in which these network models capture the dynamics of a system is not ideally suited to modelling groups of nephrons for the following reasons:

- We require classes of nodes and update rules, so we cannot model state updates as a simple propagation or percolation across the network.

- The topology of a network of nephrons remains fixed (ignoring major events such as surgical removal), so we cannot model state updates as changes to the network topology.

However, multiple graph automata can be connected together by a complex network, which can be used to capture high-level structure and hierarchy in a system. This hybrid approach allows us to separate the modelling into two distinct concerns: modelling nephrons as graph automata; and modelling groups of nephrons as complex networks of these automata.

3.3 Hierarchy in Real-World Networks

Hierarchy is important in complex systems, and in complex networks the hierarchy emerges from the network model. Explicit hierarchy is typically used when designing systems [74, 89, 202], while the hierarchy in real-world systems is not often evident and must be extracted from the model.

Most real-world complex networks exhibit four main hierarchical properties: a scale-free degree distribution [132]; small-world behaviour [186, 27]; a hierarchical structure in clustering features [150, 132]; and degree-degree correlations [131].

To extract physical knowledge from a complex system, one must focus on the right level of description for that system [66]. As a result, there is great interest in the hierarchical properties of real-world networks [115, 157] and how networks can be generated so that they exhibit these hierarchical properties [174, 26, 67]. An example of generating a network with hierarchical properties is illustrated in Fig. 7.
Figure 7: The iterative construction of a hierarchical network from a fully-connected cluster of five nodes, as demonstrated by Ravasz and Barabási in [150].

Existing studies of hierarchy in networks have focused on identifying highly clustered neighbourhoods in networks [150] and treating these neighbourhoods as networks in themselves [9]. In other words, the original system is divided into a number of modules. The definition of a module in a complex dynamical system, and the importance of inter-module dependencies, is not always unanimous nor well-quantified [187].

Song et al. demonstrate that real-world networks consist of self-repeating patterns on all scales [165]. This is unexpected, as it is widely believed that such networks are not self-similar, due to the small-world property, which implies that the relationship between the number of nodes and the diameter of the network is exponential, and not the power-law relation that is expected of self-similar networks. The self-similarity of these networks is demonstrated by calculating their fractal dimension (using a “box-counting” method). They state:

The box-counting method provides a powerful tool for further investigations of network properties because it enables a renormalization procedure, revealing that the self-similar properties and the scale-free degree distribution persist irrespectively of the amount of coarse-graining of the network.

These properties help explain why real-world networks have a scale-free distribution, and the presence of self-similarity in different networks—the World Wide Web, protein-protein interaction networks, cellular networks and social networks—suggests that this diverse range of networks may share common self-organisation dynamics.

Graph automata can also demonstrate hierarchy in behaviour. Indeed, Wolfram classifies such automata into four classes, depending on the hierarchy present in the time-state diagram [201, 6§2 p231], but little else has been done to investigate their hierarchical dynamics. Hierarchical automata do exist [74, 75, 113, 119], but are typically used to design modular systems [89] rather than to study the dynamics of existing systems. In these applications, the automata are used...
to capture the intended dynamics of the system being designed, and are not used to study any emergent dynamics that were not explicitly designed into the system. In conclusion, although hierarchical automata have been used to design modular systems, there has been little research into the emergent hierarchical properties of these automata, which is in stark contrast to the studies of hierarchy in complex networks.

3.4 DISRUPTED NETWORKS AND DISEASE

Many aspects of diseases have been studied through the use of network models. For example, the development of breast cancer has been modelled as an implicit graph automata [56] and the spread of sexually-transmitted diseases has been studied using social network models [104].

One interesting study concerning the spread of epidemics across biological, social and communication networks is that of Pastor-Satorras and Vespignani [142, 143], which examines the susceptible-infected-susceptible (SIS) model [15, 127] on complex networks. Each node of a network represents an individual system and each edge is a connection along which the infection can spread to other systems. Nodes have two available states—“healthy” (i.e., susceptible to infection) and “infected”—and at each time-step healthy nodes are infected with rate $\nu$ if they are connected to one or more infected nodes. At the same time, infected nodes are cured and again become susceptible with rate $\delta$, which defines an effective spreading rate of $\lambda = \nu / \delta$.

In networks such as random graphs, the most significant epidemiological result is the existence of a non-zero epidemic threshold $\lambda_C$ [15]. In the case $\lambda < \lambda_C$, the infection dies out exponentially fast; in the case $\lambda > \lambda_C$, the infection spreads and becomes persistent. However, over a wide range of scale-free networks Pastor-Satorras and Vespignani observed an absence of the epidemic threshold, which implies that scale-free networks are susceptible to persistent infection regardless of the spreading rate—an observation that fits well with real-world data concerning the spread of computer viruses over the Internet. Given that scale-free networks appear in a wide range of areas [4], the results obtained here may help improve our understanding of computer virus epidemics and could well have applications in other social and biological systems.

However, regardless of the scale of the models (e.g., cellular, social), the focus is on how diseases or infections spread. Our aim is to use disease modelling to investigate how the onset of renal disease in individual nephrons can affect the stability of an entire system of nephrons. To achieve this aim, the ability of diseased nephrons to affect neighbouring nephrons and the whole-system dynamics will be studied, rather than studying how the disease is propagated. With this focus on system stability, not the spread of disease, the majority of the existing disease modelling literature is not directly relevant.

However, at least one study [12] has investigated the system-wide effects of localised failure, in the context of power grids. Ash and Newth start with a simple model for representing cascading failures on complex networks (as given in e.g., [126, 50]). Each edge $(i, j)$
is assigned a weight $g_{ij}$—the efficiency of moving between $i$ and $j$ on this edge—in the range $[0,1]$. Initially, $g_{ij} = 1$ if the edge $(i, j)$ is present in the network, and $g_{ij} = 0$ if it is not.

Flow between two nodes takes the most efficient (shortest) path and the load $L_i(t)$ on node $i$ at time $t$ is equal to the number of such paths that pass through $i$ at time $t$. Each node has a finite capacity $\psi_i = L_i(0) \times \alpha$, where $\alpha$ is a measure of the stress on the system at time $t = 0$. The network is operating at maximum capacity when $\alpha = 1$, and the load on the network is light when $\alpha \gg 1$. Once the capacity of a node $i$ is exceeded, its performance is diminished as per Eq. 3.2. Initially, the capacity of some node $i$ is reduced, and at each time-step the load is redistributed over other nodes according to Eq. 3.2.

$$g_{ij}(t + 1) = \begin{cases} g_{ij}(t) \times \frac{\psi_i}{L_i(t)} & \text{if } L_i(t) > \psi_i \\ g_{ij}(0) & \text{otherwise} \end{cases} \quad (3.2)$$

The efficiency of an Erdös–Rényi random graph [57] is compared to that of a network that has been optimised using an evolutionary algorithm. For $\alpha < 1.35$ the optimised network considerably outperforms the random graph, while for $\alpha \geq 1.35$ the networks display similar behaviour. This clearly demonstrates that the optimisation algorithm has worked—network efficiency is greatly improved when the network is operating near maximum capacity and a node is impaired—and the study goes on to analyse optimised networks for topological regularities that might explain the source of the increase in resilience.

This study, unlike other network models, began with a functioning system in which the function of a single node was impaired and the resulting effect on the entire system was measured. While it may appear that this approach is identical to the study of system stability outlined in Sect. 1.1, this is not the case. This study incorporates a change in state—the reduction in capacity $\psi_i$ of some node $i$—and this change is then propagated across the network, due to the load redistribution captured in Eq. 3.2. Accordingly, the efficiency of the network is simply a weighted measure of the number of nodes whose capacities have been reduced. As such, this study is an application of percolation theory, where nodes are initially in one state (i.e., full capacity) and the percolation of another state (i.e., reduced capacity) is measured. In contrast, the approach outlined in Sect. 1.1 introduces a localised impairment to a system and observes the effect on the system function, where the impairment is contained and does not spread across the system. The contribution of an evolutionary algorithm to optimise the efficiency of a network is also unrelated to this approach, as it relies on the ability to change the topology of the system, which is not feasible in the context of renal disease.
Part III

MODELS AND EXPERIMENTS
This chapter presents the design of the model. To begin, Sect. 4.1 introduces the modelling approach—hierarchical dynamical networks—and contrasts it with other approaches. The benefits of using a discrete time model are described and a brief overview of the analysis techniques for this model is presented. The structure of the model is then described in Sect. 4.2, followed by an analysis of the structural properties of the model in Sect. 4.3. Finally, the underlying equations and update rules that capture the model dynamics are presented in Sect. 4.4.

4.1 Modelling Approach

As we have chosen to model the kidney as a complex system, the couplings between the individual parts (i.e., the tubule segments) are of more interest than the dynamics of the individual parts in isolation. To elaborate, it is of greater importance to model the major coupling and interaction mechanisms that arise in multi-nephron systems than to develop highly accurate formulations for the individual tubule segments. The kidney is known to be a remarkably resilient organ [71], capable of maintaining its function across a wide range of conditions; operating on the assumption that the kidney is indeed a complex system, our intuition is that this resilience is due to the regulatory mechanisms and other interactions that arise between the lower-level components of the kidney, rather than being primarily due to the behaviour of the individual nephron tubule segments in isolation. This approach, with less emphasis on the behaviour of individual parts than on their interactions, is similar to the approaches used in fluid dynamics and gas models, where the individual particles are essentially ignored, in favour of higher-level emergent properties such as mean flow rate and pressure. In the words of Nobel laureate Richard Feynman [77]:

We have noticed in nature that the behavior of a fluid depends very little on the nature of the individual particles in that fluid. For example, the flow of sand is very similar to the flow of water or the flow of a pile of ball bearings. We have therefore taken advantage of this fact to invent a type of imaginary particle that is especially simple for us to simulate. This particle is a perfect ball bearing that can move at a single speed in one of six directions. The flow of these particles on a large enough scale is very similar to the flow of natural fluids.

We treat the nephron tubule segments analogously—the tubule segments are modelled as highly idealised “imaginary particles” and are the building blocks of our model—and as such, they will be modelled as individual nodes in the network automata (see Sect. 3.1).
Existing network automata have typically used 2-state [205, 204], 3-state [206, 2] and even n-state [61, 118, 129] state spaces. Our state spaces are significantly more complex than [61, 118, 129] and the visual analysis methods used by existing models (and reviewed in Sect. 3.1) do not scale to the study of such complex models.

Complex systems exhibit hierarchy in their organisation [150, 187, 47, 11, 164] and the kidney is no different. Systems of nephrons exhibit hierarchy in their organisation and interact with each other in a variety of ways. Multi-nephron systems are modelled as multiple single-nephron network automata, connected to each other via an arterial tree. This tree of nephron models forms a complex network (see Sect. 3.2) but unlike the majority of existing complex network models, the hierarchy of this model will be represented explicitly in the model structure. We define the name hierarchical dynamical networks to refer to this approach of modelling a system as a hierarchical network of graph automata.

4.1.1 The effects of a discrete time model

Continuous models (in the form of systems of continuous equations) are the standard approach used in many fields, such as physics, chemistry, various engineering disciplines and commerce. The equations in these models impose constraints upon the variables in the model, which are not always directly solvable. Various numerical methods for solving such systems exist, often by converting the equations into a discrete form (and thus arriving at a discrete model). The system being modelled may in fact be discrete and the equations reflecting a more abstract view of the system—for example, although fluid dynamics are modelled using continuous equations, the fluids being modelled consist of finite numbers of molecules.

Unlike existing nephron models, the model presented here uses a discrete time model and the model dynamics are captured by systems of difference equations. When continuous systems are solved numerically—as is the case for a number of nephron models (e.g., [78])—the system is translated into discrete equations. Using this approach, solutions are calculated iteratively until a fixed point is found, which differs from the discrete time approach used here. This choice of a discrete time model has several advantages for modelling large systems, such as the kidney:

- The equations explicitly calculate the state evolution of the model, rather than capturing constraints on the state of the model.
- Coupling between equations does not affect the computational cost of solving the equations—the computational cost of the model is the sum of the computational costs of each equation.
- The system dynamics are analysed by simulating the state evolution of the model, not by obtaining a steady-state solution. Model parameters and couplings can be changed during a simulation, and the dynamic response of the system can be observed.
4.1 MODELLING APPROACH

These advantages of the discrete-time approach are important in order to meet our aims. Recall that our focus is on the nature of the interactions and couplings in multi-nephron systems. Based on the assumption that the kidney is a complex system, it is these interactions that regulate the system dynamics. By taking a network approach to our model, we can explicitly represent these interactions as edges in the network, and the use of discrete-time equations means that the equation coupling is equivalent to the explicit coupling in the network. To illustrate this point, consider three state variables $x(t)$, $y(t)$ and $z(t)$, such that:

\[
\begin{align*}
  x(t + \Delta) &= F(x(t), y(t)) \\
  y(t + \Delta) &= G(y(t), z(t)) \\
  z(t + \Delta) &= H(z(t))
\end{align*}
\]

Although $x(t)$ is indirectly coupled to $z(t)$ through $y(t)$, the value of $z(t)$ and the behaviour of $H(z(t))$ do not affect the calculation of $x(t)$. At any given time $t_i$ in a model simulation, the value of all state variables $x(t_i)$, $y(t_i)$ and $z(t_i)$ are known. Calculating $x(t_i + \Delta)$ is simply a matter of calculating $F(x(t_i), y(t_i))$, where all the necessary values are already known. The indirect coupling between $x(t)$ and $z(t)$ is realised, as the value of $z(t_i - \Delta)$ affects $y(t_i)$, which in turn affects $x(t_i + \Delta)$, but the computational cost of the whole system is simply the sum of the computational costs of the individual equations.

The choice of the time-step size $\Delta$ is important, as it determines which signals the model will be capable of reproducing [162] (illustrated in Fig. 8). It also affects the model equations, as the time-step size determines what activities occur in a single time-step. Finally, the time-step size may affect the number of time-steps that need to be simulated for the system to reach its steady-state behaviour. The choice of an appropriate value for $\Delta$ is discussed in Sect. 4.4.1.

![Figure 8: An inappropriate sampling rate can render a model incapable of reconstructing signals produced by the original system.](image-url)
When multiple parts of the model must reach an equilibrium in a single time-step, systems of coupled equations can be used to capture this explicit coupling (such an approach is used in Sect. 4.4.5). For example:

\[
x(t + \Delta), y(t + \Delta), z(t + \Delta) = F(x(t), y(t), z(t))
\] (4.4)

Simultaneous equations are generally more complex to solve than individual equations, but if the equations can be reduced to linear equations the system can be solved by Gaussian elimination. The classic algorithm uses \(O(n^3)\) arithmetic operations to solve for \(n\) equations in \(n\) variables. Improved algorithms have been developed, reaching \(O(n^{2.307})\) [171] and \(O(n^{2.376})\) [46], but these algorithms tend to have a larger constant and are only better suited to solving sufficiently large systems of equations. So while the use of simultaneous equations can increase the computational complexity of the model, the increase in complexity is bounded and can be determined easily. For small systems of simultaneous equations, the increase in complexity will have negligible impact on the complexity of the model, which is not a guarantee that can be made for small systems of coupled differential equations.

4.1.2 Analysis of discrete network automata

Perhaps the single biggest drawback of using discrete-time and difference equations is that the model does not have a “solution” of the kind generated by continuous models—instead, the dynamics of the model is determined by analysing how the model state evolves over time. In order to understand the dynamics of such a model, it must be determined when the model has reached its steady-state (which is likely to be a cycle of states rather than a single state). In general, there is no algorithm to determine how quickly a discrete model will reach a steady-state behaviour (a generalised algorithm is not possible, as it would amount to solving the Halting Problem [182]).

The standard approach is to simulate the model from an initial state and to search for patterns in the model dynamics. Typical analysis techniques for graph automata are inherently visual in nature and do not scale to large systems (see Sect. 3.1). For complex models such as the nephron model proposed here, alternative analysis techniques have to be applied, which are typically statistical in nature. Since many of the phenomena exhibited by multi-nephron systems are oscillatory in nature (e.g., filtration rates and the tubuloglomerular feedback (TGF) mechanism), signals analysis techniques such as Fourier analysis and power spectral densities will be used. Lyapunov stability measures will also be used, to determine how sensitive the model dynamics are to changes in the system. This notion of stability or sensitivity to conditions will prove particularly useful for determining how stable the model’s dynamics are when we incorporate the effect of diseases into the model (Chapter 6).

It is possible that the choice of initial state will affect the steady-state behaviour of the model (much like the choice of initial conditions
for systems of continuous equations) and the relationship between the steady-state dynamics and the choice of initial conditions must be considered in order to come to any conclusions concerning the dynamics of the model. Similarly, the stability of the model in response to changes in parameters may be harder to determine than for continuous models, as different parameter values can be used in simulations, but there is no general approach to determine the how dynamics of the model vary as a function of a parameter value (for exceedingly simple models, such an approach may be possible). Again, statistical methods will be used to measure such properties.

4.2 Model Structure

4.2.1 The single-nephron model

The nephron tubule consists of a number of functionally distinct segments—typical tubule models focus on the behaviour of a single such segment—and the single nephron model reflects this structure by treating the nephron tubule as a sequence of nodes, one for each such segment.

The nephron does not exist in isolation—the tubule sits in interstitial fluid with which it exchanges solutes (shown in Fig. 2); it is also surrounded by a network of peritubular capillaries that also exchange solutes with the interstitial fluid (shown in Fig. 1a). The single-nephron model must therefore take into account both of these structures. Note that while both the tubule and peritubular capillaries exchange solutes with the interstitial fluid, the tubule uses many active and secondary transport mechanisms to do so, while the peritubular capillaries only undergo diffusion with the interstitial fluid.

The interstitial fluid is divided into horizontal layers (as shown in Fig. 2) whose compositions differ, and so the single-nephron model reflects this layered structure. The peritubular capillaries surround the tubule, and our model divides these capillaries into segments that correspond to the tubule segments. Rather than using additional nodes to model these capillaries, each tubule segment node also models the corresponding peritubular capillaries. That is, the state of each tubule segment node encapsulates the state of the tubule segment and the state of the surrounding peritubular capillaries.

The model structure—showing fluid flow along the tubule and the TGF mechanism—is depicted in Fig. 9 and the complete network—including interstitial fluid nodes and edges for solute transport and TGF—is shown in Fig. 10.

This network captures the functional topology of a nephron, but it is not detailed enough to permit behaviours such as the counter-current exchange mechanisms in the loop of Henle. The model is refined by dividing the limbs of Henle and the medullary fluid into a number of layers (see Fig. 11) in order to model the counter-current exchange mechanism—a good example of hierarchy in a complex system. Without dividing the medullary fluid into layers a salt gradient could not exist, and without dividing the limbs into layers there would be no counter-current exchange mechanism.
Figure 9: The single-nephron model, showing tubule flow (solid arrows) and the TGF mechanism (dashed arrow). Solute transports between the tubule and the surrounding fluid are not shown.

Figure 10: The single-nephron tubule as a 1D network automata, showing the edges that capture fluid flow, solute transport and TGF.

Rather than dividing the nodes into sub-nodes, it would have been possible to model the layers as a number of state variables in the original nodes. However, this approach would have hidden the hierarchical structure of the model, preventing the model from explicitly capturing the layers and transports as nodes and edges. The number of layers into which the loop of Henle and medullary fluid are divided is a parameter of the model, allowing for control over the balance between approximating a continuous counter-current exchange mechanism and having a simple model structure.

4.2.2 The multi-nephron model

Existing models of two-nephron systems have focused on the interactions that arise due to the oscillations in filtration rate and tubule pressure [112, 166, 19, 81, 99, 19], taking into account the regulatory role played by the TGF mechanism. These models have reproduced a range of experimentally observed phenomena, such as in-phase synchronisation of both single-nephron glomerular filtration rate (SNGFR) and pressure oscillations in the proximal tubule. However, these models fail to take into account any interactions that may arise via other channels, such as changes in the interstitial fluid due to so-
Figure 11: Solute transport in the loop of Henle, demonstrating how the limbs and the interstitial fluid are divided into layers. Diffusion of $\text{H}_2\text{O}$ occurs between the descending limb and the interstitial fluid, while Na is actively transported from the ascending limbs into the interstitial fluid. The division into layers is necessary for the loop of Henle to act as a counter-current exchange mechanism. An example of typical figures for the sodium gradient in the interstitial fluid [78] are shown on the right.

lute secretion or reabsorption by a segment of the nephron tubule. The extension of the single-nephron network to a two-nephron network is designed to allow for these kinds of interactions to arise and be analysed—the network automata allows for parameters (such as interactions and couplings) to be changed during a simulation.

A two-nephron model was obtained by connecting two single-nephron models together, as shown in Fig. 12. The afferent arterioles are treated as variable resistors placed in parallel across a fixed potential (the difference between arterial and venous pressure). Thus, by changing the resistance of its afferent arteriole, a nephron can influence the SNGFR of neighbouring nephrons—this interaction is referred to as hemodynamic coupling. The afferent arterioles are assumed to be non-compliant, to simplify the arterial model.

Figure 12: The two-nephron network, showing the edges that capture fluid flow, solute transport and TGF.

The multi-nephron models are a straight-forward generalisation of the two-nephron model shown in Fig. 12, where multiple nephron...
automata share the same column of interstitial fluid (such a column is shown in Fig. 13). Again, the afferent arterioles are assumed to be non-compliant and are treated as variable resistors.

![Figure 13: A single column of interstitial fluid, showing the division of the fluid into layers. Multiple nephron automata can be placed in this column, sharing the interstitial fluid.](image)

For a small number of nephrons, it is sufficient to model the nephrons as parallel resistors placed across a potential equal to the difference between the arterial and vascular pressures. However, larger multi-nephron models must take into account the structure of the arterial tree that supplies the nephrons with blood [112].

The arterial tree for a column of interstitial fluid is modelled as a number of branches connected to a main vessel (see Fig. 14). Nephrons connected to the same branch are treated as analogous to resistors connected in parallel (with resistance equal to the resistance of the afferent arterioles) over a potential (the difference between arterial and venous blood pressure). Nephron resistance is modelled solely as the afferent arteriole resistance because the tubule and capillaries are non-compliant and therefore have a constant resistance, which we ignore for simplicity. Between each branch is a resistance which accounts for the extra distance the blood must travel to reach the lower branches.

4.2.3 The multi-column model

For sufficiently large multi-nephron models it is not reasonable for all of the nephrons to share the same interstitial fluid, as this would allow all of the nephrons in the model to directly interact with each other—something that does not occur in sufficiently large systems (such as the kidney as a whole)—and it enforces a globally-homogeneous interstitial fluid, which is not physiologically reasonable.

To separate nephrons into local neighbourhoods, and to allow the interstitial fluid to evolve inhomogeneously, the multi-column model divides the fluid into a number of columns, as shown in Fig. 15. Each column is supplied with blood from an arterial tree that has an identical structure to the arterial tree inside each column. The columns are arranged on a 2D grid, and may thus neighbour other columns.
To simulate a single time-step for this model, we simulate the time-step for each column of interstitial fluid, and then permit diffusion between neighbouring columns. We make two assumptions concern-
ing the nature of this diffusion. Firstly, we only permit horizontal diffusion (i.e., there is no diffusion between different layers of the interstitial fluid). Secondly, we only permit the diffusion to act on one quarter of the fluid in each column (see Fig. 16)—this prevents any portion of the interstitial fluid from undergoing inter-column diffusion with more than one column, as each quarter of the fluid diffuses with the neighbour on the nearest side (if any).

Figure 16: At each time-step, diffusion is performed between the fluid layers in neighbouring columns, where one quarter of the fluid in each layer is allowed to diffuse with a neighbouring layer.

Note that the arterial tree and the 2D column arrangement are entirely separate and can be structured independently of each other—e.g., columns attached to the same branch of the arterial tree need not be located near each other. As hemodynamic coupling is captured by the arterial tree and the inter-column diffusion is captured by the 2D arrangement, these two mechanisms can be controlled independently, allowing for the interactions between these mechanisms to be investigated.

4.3 Network Properties

Now that the structure of the model has been defined, it is possible to calculate several standard network properties (as defined in Sect. 3.2) that will allow the connectivity properties of models of different sizes to be compared, as well as providing a reference point for comparing the model structure to other network models and real-world networks. Note that these properties only take into account direct connections between pairs of nodes, and cannot be used to determine the likelihood of interactions between nodes that are not directly connected. All nodes are connected via paths across the network, but emergent behaviour cannot be inferred from the network topology—it must be analysed using simulations. Similarly, although nephrons can be coupled (e.g., hemodynamic coupling occurs over arterial tree), these network properties cannot infer any properties of the resulting dynamics.

The network properties of the single-column model are shown in Table 2. Not much can be inferred from the values for a single-nephron network, due to the small size of the network (14 nodes). Only when the model is extended to larger numbers of nephrons will it be possible to draw some conclusions about the network properties of the model. If we consider a two-nephron network (Fig. 12), we see that in comparison to the single-nephron network, the average degree has increased, the connection probability and the clustering coefficient have dropped, and the network diameter is unchanged.
Note that the clustering coefficient decreases for \( n < 4 \) and increases for \( n \geq 4 \). For \( n > 5 \), we have \( C > p \) (as shown in Fig. 17), which is an indication that the network is a small-world network.

<table>
<thead>
<tr>
<th>Property</th>
<th>( n = 1 )</th>
<th>( n = 2 )</th>
<th>value</th>
<th>( \lim_{n \to \infty} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Degree</td>
<td>3.83</td>
<td>4.84</td>
<td>( \frac{46n}{7n+5} )</td>
<td>6.57</td>
</tr>
<tr>
<td>Diameter</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Clustering (C)</td>
<td>0.097</td>
<td>0.065</td>
<td>( \frac{n+\frac{1}{n-1}+\frac{1}{n-3}}{7n+5} )</td>
<td>0.071</td>
</tr>
<tr>
<td>Connection Pr (p)</td>
<td>0.174</td>
<td>0.135</td>
<td>( \frac{23n}{49n^2+63n+20} )</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2: Properties of the \( n \)-nephron single-column network.

Figure 17: As \( n \) increases, the clustering coefficient (C) becomes larger than the connection probability (p).

The same network properties for multi-column models—with \( c^2 \) columns of interstitial fluid arranged in a \( c \times c \) grid, each containing \( n \) nephrons—are shown in Table 3. Note that although the network diameter tends to \( \infty \), it grows linearly with the number of columns in the model. There is essentially no difference in the mean degree, clustering coefficient or connection probability between the single-column and \( c^2 \)-column networks. In both cases the mean degree tends to \( \frac{46}{7} \approx 6.57 \); the clustering coefficient tends to \( \frac{1}{14} \approx 0.071 \); and the connection probability tends to 0.

Similar to the single-column network, we find that as the network size grows, we have \( C > p \), which again indicates that the model is a small-world network. In fact, for \( c = 1 \) we find that \( C > p \) when \( n > 5 \) (as per the single-column network) and that for \( c > 1 \) we find \( \forall n : C > p \). Both values quickly approach their asymptotic values, so that even for small values of \( n \) and \( c \) the difference between \( C \) and \( p \) is close to maximal (as shown in Fig. 18).
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>( \lim_{c \to \infty} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Degree</td>
<td>( \frac{46nc^2+12c(c-1)}{(7n+5)c^2} )</td>
<td>( \frac{46n+12}{7n+5} )</td>
</tr>
<tr>
<td>Diameter</td>
<td>( 6 + \frac{(c-1)^2}{2} )</td>
<td>( \infty )</td>
</tr>
<tr>
<td>Clustering (C)</td>
<td>( \frac{23nc^2+6c(c-1)}{(7n+5)c^2} )</td>
<td>( \frac{n}{14n+15} )</td>
</tr>
<tr>
<td>Connection Pr (p)</td>
<td>( \frac{23nc^2+6c(c-1)}{(7n+5)c^2} )</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3: Properties of the \( n \)-nephron \( c^2 \)-column network.

Figure 18: As either \( n \) or \( c \) increases, the difference between the clustering coefficient (\( C \)) and the connection probability (\( p \)) quickly approaches the asymptotic value of \( \frac{1}{14} \approx 0.071 \), as shown in this plot of \( C - p \).

4.4 State Space Equations and Update Rules

4.4.1 Choosing an appropriate time-step size

As mentioned previously (Sect. 4.1.1), the model uses discrete time. Each time-step maps the current state of the model, \( S(t) \), to the state of the model at the next time-step \( S(t + \Delta) \) (the time-step size \( \Delta \) is a parameter of the simulation). A simulation of the model is achieved by repeatedly applying this mapping, beginning with the initial model state. This mapping is realised as a number of equations, which simulate the fluid flow and solute transports in the model, and update the state variables in the model.

More importantly, the kidney exhibits multi-rate dynamics—nephron responses can range from a matter of seconds (e.g., the TGF mechanism) to the order of hours (e.g., the secretion of hormones). Specifically, the SNGFR tends to oscillate with a period of 30–40s [78, 166, 111] and the TGF signal has a delay of 3–5s [166, 145]. The model aims to recreate these signals, and therefore needs a sampling rate
capable of reproducing these signals [162]. For example, a time-step size of no greater than 1.5s is required to reconstruct a 3s delay in the TGF signal. A time-step size of one second was ultimately chosen, as this is sufficient to reconstruct signals with frequencies less than $\frac{1}{2}$Hz [162] and therefore meets our needs.

4.4.2 **Discrete fluid flow and solute transport**

Due to the use of discrete time, bodies of fluid are modelled as discrete volumes, and flow along a vessel is treated as a sequence of these discrete volumes. A vessel is able to hold a fixed number of discrete volumes, and at each time-step an entering volume of fluid forces another volume of fluid to exit the vessel (depicted in Fig. 19)—each vessel acts as a queue. In this model, all vessels (i.e., the arterial tree, the nephron tubule and the peritubular capillaries) are non-compliant, meaning their radius is fixed regardless of the hydraulic pressure of the fluid in vessel.

![Figure 19: Discrete fluid flow along a segment of tubule that contains three discrete volumes of fluid ($v_1$, $v_2$ and $v_3$). In one time-step a volume of fluid $v_0$ enters the tubule, causing the volume $v_3$ to exit the tubule.](image)

Existing models have modelled the tubule as compliant [111], modelled tubule segments as having constant resistance [19], and modelled the tubule and afferent arteriole as non-compliant [99]. Treating the tubule and capillaries as non-compliant allows the model to ignore the effects of hydrostatic pressure on resistance to fluid flow. Tubular compliance affects the speed of pulse propagation in the tubule, and the TGF time-delay can be increased to compensate for non-compliance [99].

The model does not take into hydrostatic pressure anywhere in the tubule (but hydrostatic pressure is used to determine blood flow through the arterial tree and to calculate SNGFRs). As the tubule is assumed to have no compliance, the pressure will vary in proportion to the flow rate along the tubule. The filtrate enters the tubule in discrete volumes, each of which travels along the tubule at the same rate, rather than mixing with the neighbouring fluid volumes.

The tubule segment nodes control solute transport—each node contains update rules that govern the transports between the tubule filtrate, the interstitial fluid and the peritubular capillaries. Tubule segments that have been divided into sub-nodes (i.e., the loops of Henle and the collecting ducts) perform the solute transport and fluid flow in several stages, in order to capture the dynamics of the counter-current exchange mechanism (shown in detail in Fig. 20). The stages progress as follows:
1. The fluid entering the tubule segment in the time-step is divided into sub-volumes, (the specific number is a parameter of the model).

2. A sub-volume enters the tubule segment.

3. Solute transports are simulated for each sub-node.

4. Steps 2 and 3 are repeated until all of the sub-volumes have entered the tubule segment.

Figure 20: A volume of fluid $V_0$ enters the tubule in the time-step. According to step 1, it is divided into two volumes, $V_a$ and $V_b$. $V_b$ enters the tubule, forcing $V_3$ to flow out of the tubule. Solute transports are then performed for each sub-node (represented by the superscript “1”). $V_a$ then enters the tubule, forcing $V_1^1$ to flow out of the tubule, and solute transports are performed again (represented by a superscript “2” for volumes that have undergone solute transport twice, and by a superscript “1” for $V_a$, which has undergone solute transport once).

For the sake of simplicity, the model ignores a number of factors that can affect solute transport and the composition of the tubule filtrate:

- Physiological factors such as conductivity, reflection and Michaelis-Menten kinetics.
- Chemical processes in the tubule, such as glucose consumption, lactate production and hormone secretion.
- Low-level structural features such as the various pathways between the filtrate and the interstitial fluid.
- Permeability is only taken into account for solute transports that are dependant on a hormone concentration (e.g., sodium and water reabsorption in the distal tubule and collecting ducts).
4.4.3 Model input

The input for the model consists of the blood entering the arterial tree at each time-step. The input is characterised by the concentrations of a number of solutes—the solutes used in the model are listed in Table 4. Except for water and red blood cells, all solutes are assumed to have a molarity of 0—they are not considered to contribute to fluid volume. The solute AvgProtein represents a typical protein in the bloodstream and takes its properties from albumin, the most abundant plasma protein in humans and other mammals.

The input solute concentrations are specified as ranges, having a minimum and maximum value. At each time-step, the input concentration solute is chosen at random from the range, where each value in the range has an equal likelihood of being chosen. Model parameters can also be changed at any time-step in a simulation, making the model interactive.

<table>
<thead>
<tr>
<th>Solute</th>
<th>Molar Mass</th>
<th>Has volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>18.01</td>
<td>Yes</td>
</tr>
<tr>
<td>Sodium</td>
<td>58.44</td>
<td>No</td>
</tr>
<tr>
<td>Chlorine</td>
<td>35.453</td>
<td>No</td>
</tr>
<tr>
<td>Potassium</td>
<td>39.0983</td>
<td>No</td>
</tr>
<tr>
<td>antidiuretic hormone (ADH)</td>
<td>1.5e5</td>
<td>No</td>
</tr>
<tr>
<td>Aldosterone</td>
<td>360.49</td>
<td>No</td>
</tr>
<tr>
<td>RBC</td>
<td>54.05e12</td>
<td>Yes</td>
</tr>
<tr>
<td>AvgProtein</td>
<td>6.5e4</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 4: The solutes support by the model.

4.4.4 State variables

The glomerulus and ascending thick limb nodes have internal state variables, which are used by the update rules to implement the dynamics of these tubule segments. These state variables are listed in Table 5: $R_A(t)$, $C_A(t)$, $C_E(t)$, $P_G(t)$, $\Pi_G(t)$, $\xi(t)$ and $\phi(t)$ are the state variables for the glomerulus, and $Na_M(t)$ is the state variable for the thick ascending limb. The variables $C_A(t)$ and $C_E(t)$ are calculated from the composition of the plasma at the afferent and efferent arterioles, respectively. Likewise, $Na_M(t)$ is the history of sodium delivery to the macula densa and is calculated from the composition of the filtrate leaving the thick ascending limb. The update rules for the remaining state variables are:
The glomerulus state variables, and their associated update rules, are discrete forms of the equations for the same variables in [78], while the formula for calculating oncotic pressure $\Pi(C)$ (and the associated constants $a$ and $b$) come from Deen et al. [52]. Higher-order equations for $\xi(t)$ have been used to simulate higher-frequency phenomenon [112, 166], but we have chosen to use the simpler second-order equation. The model parameters used in these update rules are listed in Table 6. The value of $D_{TGF}$ controls the time delay in the TGF mechanism, and need not be a multiple of $\Delta$—the value can be calculated by linear interpolation between time-steps.

Hemodynamic coupling arises naturally from the arterial tree network. Vascular signalling, however, does not arise naturally from the arterial tree or from the single-nephron model. To incorporate this coupling mechanism into the model, the update rule for the state variable $R_A(t)$ must be adjusted. The strength of the TGF signal is proportional to $\xi_i(t)$ for each adjacent nephron $i$ on the same branch of the arterial tree, and each such signal is assumed to have decayed to a fraction $\gamma$ of the original signal. Thus, the update rule is now:

$$\sum_{i\in N} \gamma \xi_i(t)$$

Vascular signalling has been modelled in existing models by an identical approach [112, 166].

4.4.5 State update rules

The update rules for each tubule segment node control the flow of fluid and the transport of solutes along the tubule. A summary of the function of each node is given in Table 7. Solutes are transported between the tubule filtrate and the interstitial fluid, and diffusion occurs between the interstitial fluid and the peritubular capillaries. The model does not take into account hydrostatic pressure in the interstitial fluid, so water is reabsorbed directly into the peritubular capillaries to preserve the interstitial fluid volume.

The state space for each node in the model covers the composition of the fluid in the node—$R_N$ for $N$ solutes, including water—and the node’s state variables—$R_{SV}$ for $SV$ state variables—resulting in a
<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_A(t)$</td>
<td>Afferent arteriole resistance.</td>
<td>mmHg . min / nL</td>
</tr>
<tr>
<td>$C_A(t)$</td>
<td>Afferent arteriole protein concentration.</td>
<td>g/L</td>
</tr>
<tr>
<td>$C_E(t)$</td>
<td>Efferent arteriole protein concentration.</td>
<td>g/L</td>
</tr>
<tr>
<td>$P_G(t)$</td>
<td>Hydrostatic pressure in the glomerulus.</td>
<td>mmHg</td>
</tr>
<tr>
<td>$\Pi_G(t)$</td>
<td>Oncotic pressure in the glomerulus.</td>
<td>mmHg</td>
</tr>
<tr>
<td>$\xi(t)$</td>
<td>Afferent arteriole response to TGF.</td>
<td>mmHg . min / nL</td>
</tr>
<tr>
<td>$\phi(t)$</td>
<td>The TGF signal.</td>
<td>mmHg . min / nL</td>
</tr>
<tr>
<td>$Na_M(t)$</td>
<td>Macula densa sodium delivery.</td>
<td>mol/s</td>
</tr>
</tbody>
</table>

Table 5: State variables in the nephron model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_A$</td>
<td>Arterial pressure.</td>
<td>mmHg</td>
</tr>
<tr>
<td>$P_V$</td>
<td>Venous pressure.</td>
<td>mmHg</td>
</tr>
<tr>
<td>$P_{BC}$</td>
<td>Pressure at the start of the tubule.</td>
<td>mmHg</td>
</tr>
<tr>
<td>$K_f$</td>
<td>Permeability constant for the glomerulus.</td>
<td>none</td>
</tr>
<tr>
<td>$Na_{1/2}$</td>
<td>Sodium delivery for which the TGF response is half-maximal.</td>
<td>mol/s</td>
</tr>
<tr>
<td>$k$</td>
<td>Range parameter for the TGF signal.</td>
<td>L/mmol</td>
</tr>
<tr>
<td>$D_{TGF}$</td>
<td>Time delay in sending the TGF signal.</td>
<td>s</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Natural frequency for the TGF mechanism.</td>
<td>rad/s</td>
</tr>
<tr>
<td>$\zeta$</td>
<td>Damping coefficient for the TGF mechanism.</td>
<td>none</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Viscous resistance coefficient.</td>
<td>none</td>
</tr>
<tr>
<td>$H_A$</td>
<td>The average afferent hematocrit.</td>
<td>none</td>
</tr>
</tbody>
</table>

Table 6: Model parameters.
Segment | Behaviour
--- | ---
Glomerulus | Filters a percentage of the plasma into the tubule.
Proximal Tubule | Reabsorbs $2/3$ of the tubular Na and H$_2$O.
Descending Limb | Diffuses H$_2$O to balance the Na concentrations in the tubule and the medullary fluid.
Ascending Limb | Actively transports sodium into the medullary fluid to maintain a gradient between the medullary and tubular Na concentrations.
Distal Tubule | Reabsorbs Na proportional to aldosterone concentration and reabsorbs H$_2$O proportional to ADH concentration.
Collecting Ducts | Reabsorbs Na proportional to aldosterone concentration and reabsorbs H$_2$O proportional to ADH concentration.

Table 7: The behaviour of each tubule segment in the nephron model.

state space of $\mathbb{R}^{N+S_V}$. This state space is far more complex than the typical $n$-state spaces of the cellular automata reviewed in Sect. 3.1.

For simplicity, the update rules in the model describe only the translocation of sodium and water as essential components of the filtrate. Equations for other solutes (such as chloride and potassium) can easily be incorporated into future, more complex models. Such models will also need to account for the effects of changes in filtration and reabsorption at the level of whole kidney function and the consequent modulation of extracellular fluid volume and blood pressure. In addition, the update rules take into account the effects of the hormones ADH and aldosterone, but neither hormone is secreted, reabsorbed or metabolised in the model at present.

The notation used to present the update rules is described in Table 8. The update rules are now presented.

**Glomerulus**

The role of the glomerulus node is to filter a portion of the plasma from the blood entering the nephron via the afferent arteriole into the nephron tubule. As such, this node must calculate the SNGFR for the time-step and divide the incoming blood accordingly. The equation used to calculate SNGFR is a standard equation [71, p. 317]:

$$SNGFR = K_f \times (P_G - P_{BC} - \Pi_G)$$

(4.11)

**Proximal tubule**

The proximal tubule reabsorbs 67% of the sodium and water in the tubule fluid. The water is transported directly to the capillaries (to preserve interstitial fluid volume), while the sodium is transported


<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Na}(t)$</td>
<td>Quantity of sodium at time $t$ (mol).</td>
</tr>
<tr>
<td>$\text{H}_2\text{O}(t)$</td>
<td>Volume of water at time $t$ (L).</td>
</tr>
<tr>
<td>$X(t)_I$</td>
<td>Quantity of $X$ entering the tubule segment.</td>
</tr>
<tr>
<td>$X(t)_{IF}$</td>
<td>Quantity of $X$ in the interstitial fluid that surrounds the tubule segment.</td>
</tr>
<tr>
<td>$X(t)_{I+IF}$</td>
<td>Quantity of $X$ entering the tubule segment and in the interstitial fluid that surrounds the tubule.</td>
</tr>
<tr>
<td>$X(t)_R$</td>
<td>Quantity of $X$ reabsorbed by the tubule segment.</td>
</tr>
<tr>
<td>$X(t)_D$</td>
<td>Quantity of $X$ in the descending limb.</td>
</tr>
<tr>
<td>$X(t)_A$</td>
<td>Quantity of $X$ in the ascending limb.</td>
</tr>
<tr>
<td>$X(t)_M$</td>
<td>Quantity of $X$ in the medullary fluid.</td>
</tr>
<tr>
<td>$X(t)_{D+M}$</td>
<td>Quantity of $X$ in the medullary fluid and descending limb.</td>
</tr>
<tr>
<td>$X(t)_{M+A}$</td>
<td>Quantity of $X$ in the medullary fluid and ascending limb.</td>
</tr>
<tr>
<td>$\text{Na}_G$</td>
<td>The sodium gradient maintained between the ascending limb and medullary fluid (mol/L).</td>
</tr>
</tbody>
</table>

Table 8: Notation for model equations.
into the interstitial fluid and diffused into the capillaries. The update rules are:

\[
\begin{align*}
\text{Na}(t)_R &= 0.67 \times \text{Na}(t)_I \quad (4.12) \\
\text{H}_2\text{O}(t)_R &= 0.67 \times \text{H}_2\text{O}(t)_I \quad (4.13)
\end{align*}
\]

This constant behaviour is clearly far simpler than existing proximal tubule models (such as [193] and [190]), but it is able to capture some of the essential dynamics of the tubule segment:

- The proximal reabsorption is essentially iso-osmotic [177], so sodium and water are reabsorbed in equal proportion.
- The proximal tubule normally reabsorbs about 2/3 of the filtered load of sodium and water [30, 31, 71, p. 333].

**Loop of Henle**

The major difference between the model presented here and existing models of the loop of Henle [76, 175, 108] is that in this model, the reabsorption processes are assumed to maintain an equilibrium between the filtrate and the interstitial fluid. The descending limb only permits the diffusion of water and the ascending limb only permits the active transport of sodium, as shown in Fig. 21. There are a large number of other transport mechanisms that are known to occur in the limbs (e.g., salt secretion in the descending limb [108], lactate production [76] and the recycling of urea [176]), which are currently not modelled.

Consider a horizontal slice of the loop of Henle at time \( t \), as shown in Fig. 21. At each time-step, the model simulates the diffusion of water from the descending limb and the active transport of sodium from the ascending limb. The model assumes that both the diffusion and active transport reach equilibrium during the time-step. This means that:

- Enough water is diffused to balance the ratio of sodium to water in the interstitial fluid and the descending limb.
- Enough sodium is actively transported to maintain a gradient of \( \text{Na}_G \) mol/L between the interstitial fluid and the ascending limb.

Given these assumptions and the state of slice at time \( t \), we are able to calculate the state at time \( t + \Delta \) (before the fluid flow along
the tubule is simulated). Firstly, the model conserves the total water and sodium in the system:

\[ \text{H}_2\text{O}(t)_{\text{D}+\text{M}} = \text{H}_2\text{O}(t+\Delta t)_{\text{D}+\text{M}} \]  
(4.14)

\[ \text{Na}(t)_{\text{M}+\text{A}} = \text{Na}(t+\Delta t)_{\text{M}+\text{A}} \]  
(4.15)

The descending limb reaches an equilibrium with the medullary fluid, maintaining a sodium concentration that is identical to the sodium concentration in the medullary fluid:

\[ \frac{\text{Na}(t)_{\text{M}}}{\text{H}_2\text{O}(t+\Delta t)_{\text{M}}} = \frac{\text{Na}(t)_{\text{D}}}{\text{H}_2\text{O}(t+\Delta t)_{\text{D}}} \]  
(4.16)

\[ \text{H}_2\text{O}(t+\Delta t)_{\text{D}} = \frac{\text{Na}(t)_{\text{D}}}{\text{Na}(t)_{\text{D}+\text{M}}} \times \text{H}_2\text{O}(t)_{\text{D}+\text{M}} \]  
(4.17)

\[ \text{H}_2\text{O}(t+\Delta t)_{\text{M}} = \frac{\text{Na}(t)_{\text{M}}}{\text{Na}(t)_{\text{D}+\text{M}}} \times (\text{H}_2\text{O}(t)_{\text{D}+\text{M}}) \]  
(4.18)

The ascending limb reaches an equilibrium with the medullary fluid, maintaining a sodium concentration that is \( \text{Na}_G \) less than the sodium concentration in the medullary fluid:

\[ \frac{\text{Na}(t+\Delta t)_{\text{M}}}{\text{H}_2\text{O}(t)_{\text{M}}} = \frac{\text{Na}(t+\Delta t)_{\text{A}}}{\text{H}_2\text{O}(t)_{\text{A}}} + \text{Na}_G \]  
(4.19)

\[ \text{Na}(t+\Delta t)_{\text{A}} = \frac{\text{H}_2\text{O}(t)_{\text{A}}}{\text{H}_2\text{O}(t)_{\text{M}+\text{A}}} \times (\text{Na}(t)_{\text{M}+\text{A}} - \text{Na}_G \text{H}_2\text{O}(t)_{\text{M}}) \]  
(4.20)

\[ \text{Na}(t+\Delta t)_{\text{M}} = \frac{\text{H}_2\text{O}(t)_{\text{M}}}{\text{H}_2\text{O}(t)_{\text{M}+\text{A}}} \times (\text{Na}(t)_{\text{M}+\text{A}} - \text{Na}_G \text{H}_2\text{O}(t)_{\text{A}}) \]  
(4.21)

Simulations of the two-nephron model revealed a limitation of the above equations—this system of equations assume that the loop of Henle reaches an equilibrium with the medullary fluid in isolation—the equations ignore the possibility that other loops of Henle are also interacting with the medullary fluid. When two nephrons share the same interstitial fluid, the above equations cause large oscillations in the medullary sodium concentration (see Fig. 22), as the loops of Henle both compensate for any imbalance in the equilibrium.

The solution was for the equations to ensure that each layer of interstitial fluid reached equilibrium with all of the descending and ascending limbs of Henle present in the fluid—a change that is only a generalisation of the original equations. To achieve this new equilibrium, it was necessary for the transport equations to be solved simultaneously.

Consider again a horizontal slice of the loop of Henle at time \( t \), as shown in Fig. 21. At each time-step, the model simulates the diffusion of water from the descending limb and the active transport of sodium from the ascending limb. The model assumes that both the diffusion and active transport reach equilibrium. This now means that:

- Enough water is diffused to balance the ratio of sodium to water in the interstitial fluid and all of the descending limbs.
• Enough sodium is actively transported to maintain a gradient of \( X \) mol/L between the interstitial fluid and all of the ascending limbs.

For a system of \( n \) nephrons in a single column of interstitial fluid, the equations for the loop of Henle are based on the assumption that the \( n \) nephrons cooperatively maintain the sodium gradient. Each nephron in the system reabsorbs some volume \( w_i \) of water from the descending limb \( D_i \) and some amount \( s_i \) of salt from the ascending limb \( A_i \) for each \( i \in \{1..n\} \). The state of a single layer of a loop of Henle at time \( t \) is shown in Fig. 21. At equilibrium, we obtain three equations for the water transport in the descending limbs: the conservation equations (4.22) and (4.23), and the equilibrium equation (4.24).

\[
\begin{align*}
H_2O(t + \Delta)_{D_i} &= H_2O(t)_{D_i} - w_i \quad (4.22) \\
H_2O(t + \Delta)_{M} &= H_2O(t)_{M} + \sum_i w_i \quad (4.23) \\
\frac{Na(t)_{D_i}}{H_2O(t + \Delta)_{D_i}} &= \frac{Na(t)_{M}}{H_2O(t + \Delta)_{M}} \quad (4.24)
\end{align*}
\]

Likewise, we obtain three equations for the sodium transport in the ascending limbs: the conservation equations (4.25) and (4.26), and the equilibrium equation (4.27).

\[
\begin{align*}
Na(t + \Delta)_{A_i} &= Na(t)_{A_i} - s_i \quad (4.25) \\
Na(t + \Delta)_{M} &= Na(t)_{M} + \sum_i s_i \quad (4.26) \\
\frac{Na(t + \Delta)_{M}}{H_2O(t)_{M}} &= \frac{Na(t + \Delta)_{A_i}}{H_2O(t)_{A_i}} + Na_G \quad (4.27)
\end{align*}
\]

Both systems of equations (water transport and sodium transport) can be transformed into the form \( Ax = b \), where:


\[
A = \begin{bmatrix}
1 & \alpha_1 & \cdots & \alpha_1 \\
\alpha_2 & 1 & \cdots & \alpha_2 \\
\vdots & \ddots & \ddots & \vdots \\
\alpha_N & \cdots & \alpha_N & 1
\end{bmatrix}
\]  \quad (4.28)

\[
x = \begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_N
\end{bmatrix}
\quad \text{or}
\begin{bmatrix}
s_1 \\
s_2 \\
\vdots \\
s_N
\end{bmatrix}
\quad \text{or}
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\vdots \\
\beta_N
\end{bmatrix}
\quad \text{or}
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\vdots \\
\beta_N
\end{bmatrix}
\]  \quad (4.29)

The values of \(\alpha_i\) and \(\beta_i\) depend on which system of equations we are solving. For the equations governing water reabsorption in the descending limbs, we find:

\[
\alpha_i = \frac{Na(t)_{D1}}{Na(t)_M + Na(t)_{D1}}
\]  \quad (4.30)

\[
\beta_i = \frac{Na(t)_M H_2O(t)_{D1} - Na(t)_{D1} H_2O(t)_M}{Na(t)_M + Na(t)_{D1}}
\]  \quad (4.31)

For the equations governing the active transport of sodium in the ascending limbs, we find:

\[
\alpha_i = \frac{H_2O(t)_{Ai}}{H_2O(t)_{Ai} + H_2O(t)_M}
\]  \quad (4.32)

\[
\beta_i = \frac{Na(t)_{Ai} H_2O(t)_M - Na(t)_M H_2O(t)_{Ai}}{H_2O(t)_{Ai} + H_2O(t)_M} + \frac{Na_G H_2O(t)_{Ai} H_2O(t)_M}{H_2O(t)_{Ai} + H_2O(t)_M}
\]  \quad (4.33)

The refined model solves both systems of linear equations using Gaussian elimination (with partial pivoting, to improve the stability of the solutions [179]), allowing equations 4.22, 4.23, 4.25 and 4.26 to be solved.

These equations enforce the sodium molarity at each macula densa to be equal, assuming that the macula densas are located in the same layer of interstitial fluid. This has the effect of making the TGF response of each nephron identical at every time-step, enforcing in-phase synchronisation. Experimental data, however, contradicts this [167]. A more realistic response is obtained by altering the TGF mechanism to respond to sodium delivery (mmol/s) instead of sodium concentration, an approach used by Layton et al. [99]. As a result, the values of \(Na_{1/2}\) used by the model are not directly comparable to typical values of 30–60 mmol/L [78].

Distal tubule

The first portion of the distal tubule—the early distal tubule—is similar in function to the thick ascending limb [71, p. 336], reabsorbing sodium but remaining impermeable to water. This portion of the
The distal tubule also forms part of the juxtaglomerular complex, generating the TGF signal that provides feedback control over the SNGFR. As illustrated in Fig. 9 and Fig. 10, the thick ascending limb node captures this behaviour.

The distal tubule node focuses on capturing the dynamics of the late distal tubule. This portion of the tubule has the following characteristics [71, p. 337]:

- Sodium is reabsorbed at a rate that is regulated by hormones. In particular, the reabsorption rate is proportional to the aldosterone concentration in the filtrate.

- The permeability of the tubule to water is controlled by the concentration of ADH. When ADH concentration is high, the tubule is permeable to water, but in the absence of ADH the tubule is essentially impermeable to water.

Accordingly, the update rules for the distal tubule reabsorb sodium and water in amounts that are proportional to the filtrate concentrations of aldosterone and ADH, respectively. The minimal, half-maximal and maximal tubule responses are shown in Table 9.

The update rules calculate the tubule response to ADH (Eq. 4.34) and aldosterone (Eq. 4.35)—the responses are kept within the range [0...1] (not shown in the equations)—and water and sodium are reabsorbed accordingly (Eq. 4.36 and Eq. 4.37). Note that these equations do not account for any time delay between the hormone concentrations and the according tubule response [168, 40].

\[
R_{ADH}(t) = \log_{10}(ADH(t)_{I} \times 10^{12}) \quad (4.34)
\]

\[
R_{Ald}(t) = \log_{10}(Aldosterone(t)_{I} \times 10^{11}) \quad (4.35)
\]

\[
H_{2}O(t)_{R} = R_{ADH}(t) \times \left( H_{2}O(t)_{I} - Na(t)_{I} \frac{H_{2}O(t)_{I+IF}}{Na(t)_{I+IF}} \right) \quad (4.36)
\]

\[
Na(t)_{R} = R_{Ald}(t) \times Na_{max} \times Na(t)_{I} \quad (4.37)
\]

Table 9: Distal tubule responses to hormone concentrations in the filtrate. The ADH response values are from Star et al. [168] and the aldosterone response values are from Christ et al. [40]. Note that the inhibitory effects of much higher concentrations [87] are ignored by this model.
\[ \text{Na}_R(t) = \text{volume of filtrate sodium} \times \text{tubule response} \times \text{parameter} \]

**Collecting ducts**

The cortical collecting ducts have functional characteristics similar to those of the late distal tubule [71, p. 336]. In particular, they exhibit the same reabsorption characteristics for sodium and water. The medullary collecting ducts, while differing in function from the cortical ducts in several ways, also exhibit the same reabsorption characteristics for sodium and water [71, pp. 337–8]. Accordingly, the update rules for the collecting ducts are the same as the update rules for the distal tubule—the collecting duct responses to ADH and aldosterone are calculated (Eq. 4.38 and Eq. 4.39), and water and sodium are reabsorbed accordingly (Eq. 4.40 and Eq. 4.41).

\[
\begin{align*}
R_{\text{ADH}}(t) &= \frac{\log_{10}(\text{ADH}(t) \times 10^{12})}{2} \\
R_{\text{Ald}}(t) &= \frac{\log_{10}(\text{Aldosterone}(t) \times 10^{11})}{3} \\
\text{H}_2\text{O}(t)_R &= R_{\text{ADH}}(t) \times \left( \frac{\text{H}_2\text{O}(t)_I - \text{Na}(t)_I}{\text{Na}(t)_{I+1F}} \right) \\
\text{Na}(t)_R &= R_{\text{Ald}}(t) \times \text{Na}_{\text{max}} \times \text{Na}(t)_I
\end{align*}
\]
This chapter presents an analysis of the dynamics of the model described in Chapter 4. The experimental design used to investigate the behaviour of the model was as follows:

- A single-nephron system was demonstrated to reproduce experimentally observed behaviour. Without performing this validation, the results of any experiments conducted on multi-nephron systems would be invalid. The single-nephron system was shown to exhibit regular oscillations in single-nephron glomerular filtration rate (SNGFR), due to the tubuloglomerular feedback (TGF) mechanism. It was also demonstrated that the system was capable of constructing and maintaining a stable sodium gradient in the medullary fluid, over a wide range of sodium concentrations.

- A two-nephron system was demonstrated to produce experimentally observed behaviour, consistent with existing investigations of two-nephron models. Specifically, the synchronisations that arise between the filtration rates of the two nephrons were analysed, and the effects of hemodynamic coupling and vascular signalling were shown to produce results similar to those observed in existing models. The two-nephron system also exhibited a larger, more stable sodium gradient in the medullary fluid than the single-nephron system. This is consistent with the additional concentrating power associated with multiple loops of Henle that is produced by existing models.

- The strength of the hemodynamic coupling in the two-nephron system was explored for range of values for $P_A$ and $k$. By measuring the Lyapunov stability of the two-nephron system for different values of these parameters, it was confirmed that both parameters had equivalent effects on the hemodynamic coupling in the system.

- The analysis was then extended to an 8-nephron system, which produced a larger, more stable gradient than the two-nephron system, again consistent with results obtained from existing models.

- The effects of hemodynamic coupling and vascular signalling on the 8-nephron system were then investigated. Rather than analysing the dynamics of the individual nephrons, the analysis focused on whole-system properties such as the whole-system filtration rate (WSFR). Power spectral densities were used to determine the dominant frequencies present in the WSFR. The results showed that the degree of vascular signalling present in the system not only affected the average WSFR, but also altered the dominant frequencies.

- The analysis was finally extended to determine the effects of vascular signalling on a 72-nephron multi-column system. Again,
the analysis did not focus on single-nephron dynamics, but rather on whole-column filtration rate (WCFR) and WSFR. The power spectral densities of the WCFR and WSFR again demonstrated that the degree of vascular signalling did not only affect the average filtration rates, but also affected the dominant frequencies. For example, in the absence of vascular signalling the WSFR exhibited a dominant low frequency. However, when vascular signalling was introduced, the WSFR exhibited significant higher-frequency signals.

This model is also available via the KidneyGrid portal, which lets researchers access a number of kidney models from around the world [42, 43]. The purpose of the KidneyGrid is to support collaborative research and shared access to knowledge, both within the renal science community and for external audiences (e.g., for educational purposes), without depending on single laboratories or specific programming environments [42]. Currently, KidneyGrid users are able to run simulations over ranges of parameters for two-nephron and eight-nephron systems. Visualisation is provided in the form of graphs of time-series data such as SNGFR and WSFR.

The experimental results obtained from the experiments outlined above are now presented.

5.1 THE SINGLE-NEPHRON MODEL

To validate the dynamics of the single-nephron model, the analysis focused on the regulation of SNGFR mediated by the TGF mechanism, and the ability of the loop of Henle to construct and maintain a sodium gradient in the medullary fluid. Throughout this analysis, the solute concentrations in the blood are—unless stated otherwise—as shown in Table 10.

<table>
<thead>
<tr>
<th>Solute</th>
<th>Min (mmol/L)</th>
<th>Max (mmol/L)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>SODIUM</td>
<td>135</td>
<td>145</td>
<td>[96]</td>
</tr>
<tr>
<td>CHLORIDE</td>
<td>95.0</td>
<td>103.0</td>
<td>[96]</td>
</tr>
<tr>
<td>POTASSIUM</td>
<td>3.5</td>
<td>5.0</td>
<td>[96]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Molecule</th>
<th>Min (g/L)</th>
<th>Max (g/L)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADH</td>
<td>0.0</td>
<td>4.7 × 10^{-9}</td>
<td>[1]</td>
</tr>
<tr>
<td>ALDOSTERONE</td>
<td>5.0 × 10^{-8}</td>
<td>2.5 × 10^{-7}</td>
<td>[1]</td>
</tr>
<tr>
<td>PROTEIN</td>
<td>54.0</td>
<td>54.0</td>
<td>[19]</td>
</tr>
</tbody>
</table>

Table 10: Steady-state concentration levels.

Experimental studies of SNGFR regulation (e.g., [103, 102, 78]) have observed regular oscillations in SNGFR, and tubule models support the hypothesis that these oscillations are caused by the TGF mechanism ([78, 19]).

The single-nephron model produced regular oscillations in SNGFR over a wide range of parameter values. These oscillations are driven...
5.1 The Single-Nephron Model

by the TGF mechanism, so a decrease in the strength of the feedback should result in smaller oscillations [78]. The single-nephron model produced this behaviour—plots of SNGFR for different values of the parameter \( k \) (the range parameter for the TGF signal) are shown in Fig. 23—indicating that the model has successfully captured this feature of the TGF mechanism. The dynamics of the SNGFR were not investigated further for the single-nephron model, as the purpose of the analysis was to validate the behaviour of the single-nephron model, not to exhaustively explore the range of dynamics that it could produce.

Figure 23: Regular oscillations in SNGFR.

The construction and maintenance of a solute gradient in the medullary fluid is of great importance to the regulation of water and sodium levels in the body. For example, this gradient allows the kidney to reabsorb a maximal amount of water when the body is dehydrated, producing highly concentrated urine [71, pp. 353]. This existence of this gradient is due to the ability of the loop of Henle to act as a counter-current exchange mechanism [71, pp. 352]. Accordingly, the ability of the single-nephron model to construct and maintain a gradient in the medulla was investigated.

Figure 24 shows the medullary solute gradients for the systems whose SNGFRs are shown in Fig. 23. The larger oscillations in SNGFR result in larger variations in the magnitude of the salt gradient, but all four systems demonstrate an ability to construct and maintain a stable salt gradient in the medullary fluid. The magnitude of the gradient is at least an order of magnitude smaller than the salt gradients observed experimentally—the gradient between the top and bottom of the loop can be as large as 1200mmol/L in human kidneys [169] and over 2000mmol/L in other species [158, 76]. This difference in magnitude is caused by three factors:
• It does not take into account other solutes such as chloride and urea (e.g., if chloride were reabsorbed in quantities similar to sodium, the magnitude of the gradient would be doubled);

• It does not have the concentrating power of multiple loops of Henle [97, 98, 76, 108]—this increase in concentrating power is demonstrated in Sect. 5.2; and

• It does not take into account other processes such as the production of lactate in the tubule, which may contribute significantly to accumulation of sodium [76].

Given these limitations, the ability of the model to maintain such a gradient demonstrates that the equations for the loop of Henle (Sect. 4.4) are able to capture the basic behaviour of the counter-current exchange mechanism.

![Graphs showing sodium gradient maintenance](image)

Figure 24: Construction and maintenance of a medullary sodium gradient. The graphs show the difference in sodium concentration between the outermost and innermost layers of medullary fluid.

The stability of the sodium gradient across the medullary fluid was further examined by changing the sodium delivery to the macula densa, to provoke minimal and maximal TGF responses. In order to change the sodium delivery, blood sodium levels were decreased to 110 mmol/L and increased to 170 mmol/L. These levels exceed physiologically normal levels—the body rigorously regulates the blood sodium concentration—and only serve to illustrate the range of the TGF response.

In all of the experiments shown in Fig. 24, the blood sodium concentration was 135–145 mmol/L (as listed in Table 10), and so the range of sodium delivery to the macula densa did not explore the limits of the TGF response in the model. The stability of the sodium
gradient across the medullary fluid was further examined by changing the sodium delivery to the macula densa, to provoke minimal and maximal TGF responses. In order to change the sodium delivery, two experiments were conducted. In the first experiment, blood sodium concentration was decreased to 110 mmol/L; in the second experiment, blood sodium concentration was increased to 170 mmol/L. The results of these two experiments are now presented.

For the first experiment, the blood sodium concentration was decreased by 5 mmol/L every 500 time-steps (the decreases in blood sodium concentration over time are listed in Table 11). The resulting SNGFR and medullary salt gradient are shown in Fig. 25. As the blood sodium concentration decreased, so too did the sodium delivery to the macula densa, and the TGF mechanism responded by dilating the afferent arteriole and increasing the SNGFR. As the SNGFR approached its maximal value, the size of the oscillations decreased.

Despite the decrease in blood sodium concentration, the medullary salt gradient increased by only 11% when the blood sodium concentration reached the minimal range of 110–120 mmol/L. This gradient increase arose because, although the sodium concentration in each layer of medullary fluid decreased, the sodium concentration in the top layers (the outer medulla) decreased more than the sodium concentration in the bottom layers (the inner medulla). The difference in sodium concentrations for these layers of medullary fluid are shown in Fig. 26.

In the second experiment, the blood sodium concentration was increased by 5 mmol/L every 500 time-steps—the blood sodium concentration ranges are listed in Table 12. The resulting SNGFR and medullary salt gradient are shown in Fig. 27. As the blood sodium concentration increased, so too did the sodium delivery to the macula densa, and the TGF mechanism responded by constricting the afferent arteriole and decreasing the SNGFR. As the SNGFR approached its minimal value, the size of the oscillations decreased.

Despite the increase in blood sodium concentration, the medullary salt gradient decreased by 11% when the blood sodium concentration reached the maximal range of 160–170 mmol/L. This gradient decrease arose because, although the sodium concentration in each layer of medullary fluid increased, the sodium concentration in the top layers (the outer medulla) increased less than the sodium concentration in the bottom layers (the inner medulla). The difference in sodium concentrations for these layers of medullary fluid are shown in Fig. 28.

In conclusion, this analysis of the single-nephron model dynamics demonstrate that the model is able to reproduce several patterns of behaviour that have been experimentally established. The single-nephron model exhibited regular oscillations in SNGFR whose amplitude and frequency were comparable to both experimental data and existing models. The model was also able to create and maintain a stable salt gradient in the medullary fluid, over a wider range of blood sodium concentrations than is physiologically reasonable.
<table>
<thead>
<tr>
<th>Time-step</th>
<th>Min (mmol/L)</th>
<th>Max (mmol/L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0…499</td>
<td>135</td>
<td>145</td>
</tr>
<tr>
<td>500…999</td>
<td>130</td>
<td>140</td>
</tr>
<tr>
<td>1000…1499</td>
<td>125</td>
<td>135</td>
</tr>
<tr>
<td>1500…1999</td>
<td>120</td>
<td>130</td>
</tr>
<tr>
<td>2000…2499</td>
<td>115</td>
<td>125</td>
</tr>
<tr>
<td>2500…3000</td>
<td>110</td>
<td>120</td>
</tr>
</tbody>
</table>

Table 11: Systematic decrease blood sodium concentration.

(a) The SNGFR  
(b) The medullary salt gradient

Figure 25: The single-nephron dynamics in response to decreased blood sodium concentration (k = 0.3 and PA = 110).

Figure 26: The change in sodium concentration in the top (outer medulla) and bottom (inner medulla) layers of medullary fluid, in response to decreased blood sodium concentration.
<table>
<thead>
<tr>
<th>Time-step</th>
<th>Min (mmol/L)</th>
<th>Max (mmol/L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0...499</td>
<td>135</td>
<td>145</td>
</tr>
<tr>
<td>500...999</td>
<td>140</td>
<td>150</td>
</tr>
<tr>
<td>1000...1499</td>
<td>145</td>
<td>155</td>
</tr>
<tr>
<td>1500...1999</td>
<td>150</td>
<td>160</td>
</tr>
<tr>
<td>2000...2499</td>
<td>155</td>
<td>165</td>
</tr>
<tr>
<td>2500...3000</td>
<td>160</td>
<td>170</td>
</tr>
</tbody>
</table>

Table 12: Systematic increase in blood sodium concentration.

Figure 27: The single-nephron dynamics in response to increased blood sodium concentration ($k = 0.3$ and $P_A = 110$).

Figure 28: The change in sodium concentration in the top (outer medulla) and bottom (inner medulla) layers of medullary fluid, in response to increased blood sodium concentration.
5.2 TWO-NEPHRON SYSTEMS

With the single-nephron model dynamics validated, the focus of the analysis turns to multi-nephron systems, which allow the interactions and couplings between nephrons to be investigated. The multi-nephron analysis begins with the simplest multi-nephron system—a two-nephron model, where each nephron has an identical structure and location in the renal fluid layers, and vascular signalling is absent ($\gamma = 0$).

The first experiment investigated the effects that changing the arterial pressure ($P_A$) and the steepness of the TGF response ($k$) have on the SNGFRs of the two nephrons. The state-space of $P_A \times k$ was explored for values of $P_A$ ranging from a normal level of 108.5 mmHg [78] to a maximum of 114 mmHg in 0.5 mmHg increments, and for values of $k$ ranging from 0.25 to 0.4 [78] in increments of 0.01. As $P_A$ or $k$ was increased, the two nephrons were observed to change from in-phase synchronisation to anti-phase synchronisation, with a region in the middle where the nephrons were neither in-phase or anti-phase. This trend is summarised in Fig. 29, which shows the mean absolute difference in filtration rates of the two nephrons over the parameter space. This demonstrates the increasing the arterial pressure ($P_A$) and increasing the steepness of the TGF response ($k$) have similar effects on the two-nephron system, as suggested in [19].

Figure 29: Mean difference in filtration rates in a two-nephron system, over the parameter space ($P_A$, $k$).

To measure the stability of the model in response to these parameter changes, Lyapunov exponents were calculated about several points in the phase space. An exponent $L_y$ for point $y$ around point $x$ indicates that the behaviour of the system at $y$ approaches the behaviour of the system at $x$ no slower than $\exp^{L_y}$. The negative Lyapunov exponents (i.e., those where the behaviours of the two systems converge) about each of the chosen points are shown in Fig. 30. These graphs show that the system exhibits stable behaviour along the lines $P_A - k = C$, for some fixed $C$. This further reinforces the
hypothesis that $P_A$ and $k$ have equivalent effects on the system [19], as an increase of one can be countered by a decrease in the other, and vice versa.

The behaviour of the model was also examined a wider range of arterial pressures $P_A$ (80–130 mmHg). At high pressures (115–130 mmHg) the model exhibited anti-phase synchronisations, and due to the non-compliant nature of the model, the difference between minimal and maximal filtration rates rose to 23 nL/min at $P_A = 130$—shown in Fig. 31a. At low pressures (80–100 mmHg) the model exhibited in-phase synchronisations, and the difference between minimal and maximal filtration rates decreased to 2.5 nL/min at $P_A = 80$ mmHg, again due to the non-compliant nature of the model—shown in Fig. 31b.
Anti-phase synchronisations have not been observed experimentally [167], as the vascular coupling between nephrons in the kidney overwhelms the hemodynamic coupling [112]. The two-nephron model has been shown to produce anti-phase synchronisations in the absence of vascular coupling, and it will now be shown that incorporating vascular coupling into the model is sufficient to prevent anti-phase synchronisations from arising, which is consistent with the results presented by Marsh et al. [112] and further validates the model.

As hemodynamic and vascular coupling reinforce opposing behaviours, the two-nephron model is a suitable candidate for analysing how the combination of these opposing mechanisms affect nephron behaviour. The analysis so far has been conducted with hemodynamic coupling present in the model (via the vascular network) and without any vascular coupling ($\gamma = 0$). As expected, the hemodynamic coupling enforced anti-phase synchronisation between the two filtration rates when the coupling was sufficiently strong (see Fig. 29).

Accordingly, the two-nephron system exhibited regular anti-phase oscillations for a range of $P_A$ and $k$ values. To investigate the ability of vascular coupling to prevent anti-phase synchronisation, the values $P_A = 110.5$ mmHg and $k = 0.3$ were chosen, and the SNGFRs of the two nephrons were analysed over a range of $\gamma$ values. The results are shown in Fig. 32, which demonstrates that as $\gamma$ is increased, the anti-phase nature of the SNGFR oscillations is decreased until the system exhibits in-phase synchronisation. The magnitude of the oscillations also decrease as $\gamma$ is increased, since vascular signalling causes the TGF signals to each afferent arteriole to approach the same value (i.e., when $\gamma = 1$ the signals will be identical).

If the strength of the hemodynamic coupling is increased—that is, if one or both of $P_A$ and $k$ are increased—in the presence of vascular signalling, then the anti-phase synchronisation becomes more prevalent. This is illustrated in Fig. 33, where $k$ has been increased from 0.2 to 0.3. Under these conditions, the effect of $\gamma = 0.010$ is similar to the effect of $\gamma = 0.005$ when $k = 0.2$ (compare Fig. 32a and Fig. 33a). Similarly, for the case of $\gamma = 0.11$ the dynamics are similar to the effect of $\gamma = 0.008$ when $k = 0.2$ (compare Fig. 32b and Fig. 33b).

With two identical nephrons, the coupling mechanisms compete to force in-phase and anti-phase synchronisations on the system. If some asymmetry is introduced into the model (e.g., if the time delay
in one of the TGF mechanisms is altered), then the TGF responses of the two nephrons will be driven to different frequencies. The effect of such an asymmetry was investigated and the results are shown in Fig. 34, where an additional two second delay was added to the TGF mechanism of one of the nephrons.

As was the case for two identical nephrons, without any vascular coupling the hemodynamic coupling forces an anti-phase synchronisation between the two SNGFRs, despite the different time delays in the two TGF mechanisms. As the vascular coupling is increased, the magnitude of the oscillations decrease and become less regular—as
Figure 34: The effect of varying the strength of vascular coupling between two nephrons, whose TGF delays differ by two seconds.
shown in Fig. 34d for $\gamma = 0.08$—in the same manner as for the system of two identical nephrons. The two SNGFRs approach in-phase synchronisation as the vascular coupling is further increased. However, unlike the system of two identical nephrons, some differences between the two SNGFRs remain even when $\gamma = 0.20$ (shown in Fig. 34j). Thus, the different delays in the TGF mechanism prevent the two nephrons from exhibiting identical SNGFRs.

Now that the interactions between the hemodynamic and vascular coupling mechanisms have been investigated, the analysis turns to the medullary salt gradient produced by the two-nephron model. When the blood sodium concentration was within the normal range of 135–145 mmol/L, the two-nephron model was able to maintain a stable salt gradient 46% larger than the gradient produced by the single-nephron model, as shown in Fig. 35. This increase in the salt gradient is due to the increased concentrating power of multiple loops of Henle.

The two-nephron system also produced a more stable salt gradient than the single-nephron system—Fig. 36 shows the salt gradient when blood sodium concentration was decreased as per Table 11, and Fig. 37 shows the salt gradient when the blood sodium concentration was increased as per Table 12. In response to these changes in blood sodium concentration, the salt gradient in the two-nephron system exhibited smaller variations than the single-nephron system. These variations are summarised in Table 13. This result suggests that adding more nephrons to the model results in a larger, more stable gradient, which is consistent with existing medullary models [97, 98, 76, 108].

5.3 AN INHOMOGENEOUS 8-NEPHRON SYSTEM

The two-nephron system was used to analyse the interaction between the competing coupling mechanisms of hemodynamic coupling and vascular signalling—and also to investigate the stability of the medullary salt gradient maintained by multiple loops of Henle—in the simplest possible system. The analysis is now extended to an eight-nephron system that consists of four nephron pairs, each pair having longer loops of Henle than the last. The structure of this system is shown in Fig. 38.

The medullary salt gradient of the system was investigated over the same changes in blood sodium concentration as the single-nephron and two-nephron systems (the changes are listed in Table 11 and Table 12). In comparison to these systems, the 8-nephron system maintained a larger, more stable medullary gradient, as illustrated in Fig. 39. The sodium concentrations at the top (outer medulla) and bottom (inner medulla) of the medullary fluid are shown in Fig. 40, and the oscillations in these concentrations are much smaller than those in the previous systems (e.g., compare to Fig. 36 and Fig. 37). This is further evidence that the combined dynamics of multiple loops of Henle result in a larger medullary gradient, which is again consistent with existing medullary models [97, 98, 76, 108]. As shown in Fig. 41, in response to these changes in blood sodium concentration
Figure 35: The medullary sodium gradient in a two-nephron system when blood sodium concentration remains at 135–145 mmol/L.

Figure 36: The medullary sodium gradient in a two-nephron system when blood sodium concentration progressively drops to 110–120 mmol/L.

Figure 37: The medullary sodium gradient in a two-nephron system when blood sodium concentration progressively rises to 160–170 mmol/L.
5.3 An inhomogeneous 8-nephron system

the WSFR increases and decreases in the same manner as the SNGFR in the single-nephron and two-nephron systems.

Note that while this 8-nephron system has exhibited a larger medullary salt gradient than the previous single-nephron and two-nephron systems, the increase is relatively small—21%—when compared to the increase of 46% that was observed between the single-nephron and two-nephron systems. The likely causes for this are: the 8-nephron system may have approached the maximal medullary gradient that can be produced by the equations in Sect. 4.4.5; and the value of $Na_{1/2}$ (see Table 6) is the same for each nephron, which may be unreasonable as the sodium delivery to the macula densa will differ depending on the length of the loop of Henle.

While the two-nephron system was used to investigate the effects of vascular signalling on the synchronisations between individual nephrons, the 8-nephron system was used to investigate the effects of vascular signalling on the WSFR of a system of inhomogeneous nephrons. Figure 42 shows the WSFR of the 8-nephron system for several values of $\gamma$, the percentage of the TGF signal that leaks to neighbouring nephrons. These simulations demonstrated that the presence ofvascular signalling decreased the average WSFR and also led to smaller oscillations in WSFR.

These trends are not surprising, since vascular signalling increases the TGF signal sent to each nephron—the nephron receives a fraction of neighbouring TGF signals in addition to its own TGF signal—which leads to a decrease in the SNGFR of each nephron. And as shown in Sect. 5.1, the size of the oscillations in SNGFR are decreased when the sodium delivery to the macula densa differs greatly from the half-maximal value $Na_{1/2}$ (see Fig. 3). Accordingly, increasing the degree of vascular signalling in the 8-nephron system decreases the average SNGFR and the size of the oscillations in SNGFR, which results in similar effects on the WSFR.

However, the power spectral densities of the different WSFRs (shown in Fig. 43) demonstrate that the degree of vascular signalling has a more complex effect on WSFR than simply reducing the average WSFR and the magnitude of WSFR oscillations. The different peaks exhibited by the power spectral densities indicate that different dominant frequencies are present in the WSFR, depending on the value of $\gamma$. This means that the vascular signalling has affected the SNGFR frequencies of the individual nephrons in the system, due to vascular signalling.

Figure 38: Structure of the 8-nephron model.
Sodium (mmol/L) | Single nephron | Two nephrons
---|---|---
110–120 | +9.8% | +0.6%
160–170 | −13.0% | −7.5%

Table 13: The difference in the sodium gradient at standard (135–145 mmol/L) and altered blood sodium concentrations, which demonstrates that the two-nephron system maintains a more stable gradient than the single-nephron system.

Figure 39: The medullary salt gradient in an eight-nephron system when blood sodium concentration is altered.

Figure 40: The salt concentration in the inner and outer medulla of an eight-nephron system when blood sodium concentration is altered.

Figure 41: The WSFR of an eight-nephron system when blood sodium concentration is altered.
Figure 42: The WSFR of eight-nephron systems with different levels of vascular signalling.
Figure 43: The WSFR power spectral density of eight-nephron systems with different levels of vascular signalling.
between nephrons whose loops of Henle have different lengths. More interestingly, the spectral densities for $\gamma = 0.000$ (Fig. 43a) and $\gamma = 0.005$ (Fig. 43f) are very similar, despite the spectral densities for $0.000 < \gamma < 0.005$ showing no such similarity. This suggests that the WSFRs shown in Fig. 42a and Fig. 42f have similar properties—despite the obvious differences in average and oscillation size—which is a surprising observation.

5.4 A 72-NEPHRON MULTI-COLUMN MODEL

The eight-nephron system was used to analyse the interaction between the competing coupling mechanisms of hemodynamic coupling and vascular signalling—and also to investigate the stability of the medullary salt gradient maintained by multiple loops of Henle—in a system of inhomogeneous nephrons. The analysis is now extended to a 72-nephron system consisting of nine columns arranged in a $3 \times 3$ grid, with each column having a structure identical to that of the eight-nephron system presented in Sect. 5.3. The 2D arrangement of the columns and the structure of the arterial tree are shown in Fig. 44.

![Diagram](image)

(a) The 2D arrangement of interstitial fluid columns. (b) The arterial tree connecting the columns.

Figure 44: The high-level structure of the 72-nephron model. Each column is assigned a number 1–9. The internal structure of each column is identical, and is shown in Fig. 38.

In the absence of vascular signalling (i.e., $\gamma = 0$), the WSFR exhibits oscillations of 100–200 nL/min (see Fig. 45a). The power spectral density shows that these oscillations predominantly occur at a single frequency—the frequency at which the longest-looped nephrons naturally oscillate. This is different to the power spectral density of the 8-nephron system, where higher-frequency peaks are present (see Fig. 43a), and indicates that the majority of higher-frequency signals (such as the oscillations of the shorter-looped nephrons) cancel each other out in this system. Figure 46 shows the individual WCFRs, and the power spectral densities clearly indicate the presence
of higher-frequency oscillations in the WCFRs. Note that although the structure of the individual columns are homogeneous, the WCFR plots and power spectral density plots differ, which indicates that the higher-level structure of the system (i.e., the 2D column arrangement and the arterial tree connecting the columns) is affecting the dynamics of each column.

Once vascular signalling is introduced into the 72-nephron system, the patterns in the WSFR become less clear. As shown in Fig. 47, the addition of vascular signalling results in higher-frequency signals becoming more dominant than when \( \gamma = 0 \), although this trend is not entirely consistent—witness Fig. 47c. The individual WCFRs, however, show a consistent increase in the presence of higher-frequency signals when compared to their behaviour when \( \gamma = 0 \)—compare the power spectral densities in Fig. 48 with those in Fig. 46.

Finally, it was observed that as the degree of vascular signalling was increased, the average WSFR and WCFR decreased—shown in Fig. 49—which is consistent with the results produced by the 8-nephron system in Sect. 5.3.

### 5.5 Summary of Experiments

The experiments in this chapter have demonstrated that the model presented in Chapter 4 produces valid dynamics across a range of multi-nephron systems.

Firstly, a single-nephron system was shown to produce oscillations in SNGFR mediated by the TGF mechanism, and was also shown to maintain a stable salt gradient in the medullary fluid across the physiological range of blood sodium concentrations.

A two-nephron system was then used to investigate the interactions that arise between nephrons. Over the state space \( k \times P_A \), the system was shown to transition between in-phase and anti-phase oscillations in the absence of vascular signalling, and the parameters \( k \) and \( P_A \) were shown to have equivalent effects on these oscillations. When vascular signalling was introduced to the system, it was shown to overwhelm the hemodynamic coupling present in the system and enforce in-phase oscillations, even when the nephrons featured different delays in the TGF mechanism. This system also maintained a larger, more stable medullary salt gradient than the single-nephron system, suggesting that the presence of multiple loops of Henle is responsible for the stability and magnitude of the gradient.

An eight nephron system was then demonstrated to maintain a medullary salt gradient that was both larger and more stable than the two-nephron system, providing further evidence that additional loops of Henle lead to larger, more stable medullary salt gradients. The effects of vascular signalling on the WSFR were then investigated, and it was shown that increased levels of vascular signalling decreased the average WSFR. Moreover, the spectral densities of the WSFR demonstrated that the degree of vascular signalling affected the dominant frequencies in the WSFR.

Finally, a 72-nephron system of 9 columns was analysed. Although the columns were homogeneous, the WCFRs differed, depending on which branch of the arterial tree the column was connected to. An
(a) The WSFR.  
(b) The WSFR power spectral density.  

Figure 45: The WSFR of the 72-nephron system when $\gamma = 0$.

(a) The WCFR of columns 1–3.  
(b) The spectral density of columns 1–3.  

(c) The WCFR of columns 4–6.  
(d) The spectral density of columns 4–6.  

(e) The WCFR of columns 7–9.  
(f) The spectral density of columns 7–9.  

Figure 46: The individual WCFRs in the 72-nephron system when $\gamma = 0$.  

5.5 SUMMARY OF EXPERIMENTS
Figure 47: Power spectral densities of the WSFR of the 72-nephron system for different values of $\gamma$. 
(a) Columns 1–3 when $\gamma = 0.003$. (b) Columns 1–3 when $\gamma = 0.005$.

(c) Columns 4–6 when $\gamma = 0.003$. (d) Columns 4–6 when $\gamma = 0.005$.

(e) Columns 7–9 when $\gamma = 0.003$. (f) Columns 7–9 when $\gamma = 0.005$.

Figure 48: Power spectral densities of the WCFRs of the 72-nephron system for different values of $\gamma$.

(a) The change in average WSFR. (b) The change in average WCFR.

Figure 49: Average WCFRs and WSFRs of the 72-nephron system for different values of $\gamma$. 
analysis of the power spectral densities showed the arterial tree also influenced the dominant frequencies in the WCFRs. Increasing the degree of vascular signalling in this system was shown to result in the presence of higher-frequency signals in the WCFRs and WSFR, and also to reduce the average WCFRs and WSFR.
The dynamics of several multi-nephron systems have been analysed and presented in Chapter 5, providing a benchmark against which the behaviour of similar systems can be compared. In particular, such benchmarks allow the system-wide effects of localised changes in dynamics (hereafter referred to as impairments) to be determined. The magnitude of the effect of an impairment on the whole-system dynamics forms a measure of the stability or resilience of the system.

The most obvious application of this approach—particularly useful when applied to a whole-kidney model—is to model renal diseases as impairments and to study how the diseases affect the whole-system dynamics. Such studies could investigate which symptoms of a renal disease have the most significant impact on renal function, or determine what fraction of the nephron population must become impaired to reach a particular stage of chronic kidney disease (CKD). The most common symptoms of renal disease have been introduced in Sect. 2.3, and the application of network models and automata to studying disease has been reviewed in Sect. 3.4.

This chapter presents an example of how to incorporate an impairment into the model and how to determine the impact of the impairment on the dynamics of multiple systems and at different levels of hierarchy. This example introduces a simple impairment that attempts to capture a single effect of proteinuria—the presence of abnormally large amounts of protein in the urine (see Sect. 2.3 for further details). This impairment only serves to illustrate the general approach, and not to predict the effects that proteinuria might have on kidney behaviour.

To begin, the method by which impairments are incorporated into the model is described, and the control that the modeller has over the incidence and spread of an impairment is discussed. Secondly, the choice of the specific impairment is justified, and the design of the impairment is presented. Thirdly, several techniques for measuring the effects of the impairment on the system dynamics are outlined, focusing on the dynamics at several levels of hierarchy.

The dynamics of several impaired systems are then presented and compared to the dynamics of the original (unimpaired) systems. Two experiments investigate the effects that a single impaired nephron and pairs of impaired nephrons have on an 8-nephron system; two further experiments investigate the effect of impairments in the context of a 72-nephron system. The chapter concludes by drawing some conclusions about the stability of the model in response to the chosen impairment.

6.1 INCORPORATING IMPAIRMENTS INTO THE MODEL

For an impairment to change the model dynamics, it must change either the model structure (Sect. 4.2) or the underlying equations
EMERGENT EFFECTS OF LOCALISED IMPAIRMENT

(Sect. 4.4). The high-level structure of the model (Fig. 14 and Fig. 15) captures the physical structure of the kidney, which should not be altered by an impairment. The lower-level structure (Fig. 12) captures the functional structure of the nephrons, which is closely tied to the physical structure of the nephrons and should also not be altered by an impairment. Instead, it is clear that impairments should only introduce changes to the underlying model equations.

To allow for both impaired and unimpaired nephrons to exist in a multi-nephron system, it is necessary for each nephron to include an impair state that indicates whether the nephron is impaired or not. When the model equations are applied to a particular nephron, the impair state is examined and the appropriate modifications to the model equations are applied. It is straightforward to extend this approach to support multiple impairments.

Once the impairments have been defined, the evolution of the nephron impair states over time must be specified. All nephrons in a model are initially unimpaired and thus exhibit normal behaviour. The different ways in which a nephron may become impaired are now described.

6.1.1 Stochastic impairments

Each impairment is assigned a failure probability, which defines the likelihood that a nephron will become impaired over a single time-step. This probability can be defined with respect to time (i.e., a fixed probability for each time-step) or with respect to blood flow to the nephron at that time-step (i.e., a probability that can vary with each time-step).

Similarly, each impaired is assigned a recovery probability, which defines the likelihood that an impaired nephron will become unimpaired. This probability can also be defined with respect to time or to blood flow to the nephron.

Note that this approach models the transition between impair states as a Markov Chain.

6.1.2 Stochastic spread of impairments

Impairments are also assigned a spreading probability that defines the likelihood of that impairment spreading from an impaired nephron to a neighbouring nephron over a single time-step. For this to be well-defined, a definition of what constitutes a neighbouring nephron must be given.

Nephrons in a single column of interstitial fluid are arranged in a number of branches (see Sect. 4.2.2 and Fig. 14). If two nephrons in the same branch are next to each other, it is natural to consider them as neighbours. The first nephron in a branch is also considered to be a neighbour of the first nephron in the adjacent branches above and below it (again, see Fig. 14). More concisely, nephrons are neighbours if they are adjacent in either a depth-first or breadth-first traversal of the column’s nephron tree. Finally, the first nephron in an entire column is considered to be a neighbour of the first nephrons in adjacent columns (see Fig. 15 and Fig. 16).
6.1.3 Deterministic impairments

By allowing impairments to arise, spread and recover in a stochastic manner, it can be observed how they spread across the model, in a similar manner to the network models used in epidemiology (see Sect. 3.4). However, studying the effects that impaired nephrons have on a system of nephrons— as opposed to studying the manner in which the impairments spread— requires explicit control over the impair state of the nephrons.

To this end the model also supports deterministic impairments, where any number of impairments can be introduced to particular nephrons in a system at specific time-steps. These impairments occur in parallel with the stochastic methods described previously, and for absolute control over the impair states it is necessary to set the failure, spreading and recovery probabilities of the impairment to zero.

6.2 Choice of impairment

As outlined in Sect. 2.3, there are many diseases that affect the kidney. For the purposes of disease modelling, however; only the actual symptoms of the disease are of interest—the specific causes (e.g., buildup of scar tissue, calcium deposits) are not directly relevant. As such, the modelling concern does not focus on a specific renal disease, but rather on the symptoms of that disease. By designing impairments for these symptoms, the effects of renal diseases can be modelled as specific combinations of these impairments.

As was stated in Sect. 2.3, diabetic nephropathy, hypertension and glomerulonephritis account for around 75% of all adult cases of chronic renal failure (CRF) [41], with diabetic nephropathy the most common cause of CRF and end-stage renal failure (ESRF) in the USA [53]. The focus of the impairment example to be presented here is therefore on the symptoms presented by these diseases:

**EDEMA** Excess fluid in the body.

**HEMATURIA** Blood present in the urine.

**HYPERLIPIDEMIA** Raised levels of lipids in the blood.

**HYPERTENSION** Chronically elevated blood pressure.

**HYPOALBUMINEMIA** Abnormally low albumin levels in the blood.

**PROTEINURIA** Protein present in the urine.

Some of these symptoms are caused by the presence of other symptoms— e.g., proteinuria can result in hypoalbuminemia, which reduces the oncotic pressure in the blood and can lead to edema—and are not relevant in isolation. Other symptoms can be caused by changing model parameters— e.g., hypertension can be simulated by increasing the arterial pressure parameter— having a system-wide effect as opposed to the per-nephron effects of impairments in our model, and must be controlled separately. Of the symptoms
described here, the impairment that will now be presented only considers proteinuria—this impairment serves only as an example of this approach, and not as a complete, detailed model of one or more symptoms.

6.2.1 Implementation of the impairment

Proteinuria is the presence of abnormally large amounts of protein in the urine. As proteins are not secreted by the tubule, this condition arises due to the presence of proteins in the tubule filtrate. While this can have a number of effects on the function of the tubule, the only effect that will be captured by the impairment is that on the reabsorption of water in the descending limb of Henle.

The presence of protein in the filtrate increases the oncotic pressure of the filtrate and thus retards the diffusion of water from the filtrate into the surrounding interstitial fluid. Accordingly, the impairment will capture this reduction in reabsorption by modifying the equations that govern the reabsorption of water in the limbs of Henle (as given in Sect. 4.4.5).

Specifically, for each descending limb that is affected by the impairment, the amount of water reabsorbed into the interstitial fluid is some fraction $D_i$ of the amount that would be reabsorbed were the tubule not affected by proteinuria. This is modelled by changing the $w_i$ coefficients in Eq. 4.29 to take into account $D_i$, resulting in Eq. 6.1. For impaired nephrons, the value of $D_i$ is a model parameter, while for unimpaired nephrons $D_i = 1$.

$$x = \begin{bmatrix} w'_1 \\ w'_2 \\ \vdots \\ w'_N \end{bmatrix}, \quad w'_i = \frac{w_i}{D_i}$$  \hspace{1cm} (6.1)

6.3 Measuring the Effects of Impairments

The goal of this example impairment is to determine how system dynamics are affected by impaired nephrons, and so the focus of the analysis will be on whole-system dynamics rather than the dynamics of individual tubule segments.

Existing multi-nephron models have focused on single-nephron glomerular filtration rate (SNGFR) [99] and tubular pressure [112, 81]; due to the design of the model, pressure is not calculated anywhere in the tubule. Pressure can be treated as varying in proportion to flow rate, if the tubule is assumed to be non-compliant or only slightly compliant. This point is discussed further in Sect. 4.4.2.

In the experiments performed previously (Chapter 5), the analysis of the system dynamics focused on SNGFR—in particular, the frequency of the oscillations and the phase (relative to other nephrons)—and so SNGFR will form the starting point of the analysis here. With the emphasis on whole-system dynamics, the analysis will also extend to
comparing the whole-system filtration rate (WSFR) of unimpaired and impaired systems.

Based on the assumption that the filtration rate at any level of hierarchy (e.g., single nephrons, multiple columns) can be characterised by a small number of measurements (e.g., frequency, amplitude, phase and mean value), the low-level dynamics can be ignored. Instead, the analysis can determine the effects of the impairment—and the degree to which the effects permeate across the system—by comparing these characteristic measurements in the impaired and unimpaired systems at multiple levels of hierarchy.

Once these effects have been analysed and quantified, it may transpire that the impairment can be simplified—i.e., some of the changes to the model equations may be removed without affecting the system dynamics—allowing the modeller to determine which aspects of the impairment have the largest effects. To draw an analogy with flocks of birds [152, 153]—where three exceedingly simple rules for individual navigation are sufficient to generate realistic aggregate motion for an entire flock—a thorough, detailed set of impairments may prove to be more complex than necessary to model a chosen phenomena, such as a particular renal disease.

6.4 Impairments in an 8-Nephron System

The first investigation was to analyse the effect that impaired nephrons have on the dynamics of the eight-nephron system presented in Sect. 5.3. This system consists of four nephron pairs, each pair having longer loops of Henle than the last—this arrangement is shown in Fig. 50.

![Figure 50: The loops of Henle in the eight-nephron system. Each nephron is identified by a number from 1 to 8, and the length of the loops are rated on a scale of 0 to 3.](image)

Two separate experiments were conducted, to investigate the effects of a single impaired nephron and the effects of two impaired nephrons. These experiments are now presented and compared, and the effects of impaired nephrons on the eight-nephron system are summarised.

In the following discussion, nephrons are frequently referred to by the length of their loop of Henle (0 . . . 3) and their nephron number (1 . . . 8). The SNGFR oscillation frequency of the nephron pair with loop of Henle length L is also referred to as the natural frequency of loop length L, or the loop length L frequency.
6.4.1 A single impaired nephron

The first experiment was to investigate how a single impaired nephron could affect the eight-nephron system. For each nephron in the system, simulations were run where the nephron was deterministically infected at $t = 1,200$ and the simulation ran until $t = 3,000$. A simulation was run for each value of $D_P$ from 0.5 to 0.95—in steps of 0.05—and the model parameters and conditions were otherwise identical to the experiments performed upon the system in Sect. 5.3.

A comparison of the WSFR for each of these simulations is presented in Fig. 51. Fig. 51a shows the average WSFRs and Fig. 51b shows the magnitude of the oscillations in WSFRs. From Fig. 51a it is clear that the impairment decreased the average WSFR, and that the magnitude of this decrease is proportional to both $D_P$ and the length of the impaired nephron’s loop of Henle. However, Fig. 51b shows that the oscillations in WSFR were not affected by the impairment unless one of the longest-loop nephrons was impaired and $D_P$ was small ($\leq 0.6$).

A more detailed comparison between the original eight-nephron system and one of the impaired systems will now be given. Fig. 51 shows that the impaired system that differed the most from the original system was when one of the longest-loop nephron was infected and $D_P = 0.5$, and so it is with this impaired system that the original system will be compared. Specifically, in this impaired system the impaired nephron is nephron 7.

The WSFR of the two systems are shown in Fig. 52a, where the decrease in average WSFR due to the impairment can clearly be seen. The power spectral densities of the two WSFRs are shown in Fig. 52b, and the impaired system can be seen to have a more pronounced peak at a low frequency. This suggests that the SNGFRs of the longer-looped nephrons are experiencing larger oscillations in the impaired system than in the original system.

The SNGFRs of the individual nephrons in the impaired system will now be examined. The power spectral densities of the SNGFRs are presented in Fig. 53. This figure shows that while nephrons 3–8 are essentially oscillating at a single frequency, nephrons 1 and 2 (the nephrons with the shortest loops) are oscillating at a combination of three different frequencies. From the other power spectral densities, it is apparent that the spikes in the power spectral densities of nephrons 1 and 2 correspond to the natural frequencies of the loop lengths 0, 1 and 3—there does not appear to be a spike at the natural frequency of loop length 2. The frequency of the oscillations in SNGFR is clearly proportional to the length of the loops of Henle, as expected (see Sect. 5.3). Note that the impaired nephron (nephron 7) has a larger spike than nephron 8, while the other nephron pairs exhibit power spectral densities that are identical.

To compare these spectral densities to those of the same nephrons in the original system, plots of the differences in spectral densities between the original and impaired systems are given in Fig. 54. It is evident that nephrons 7 and 8 have larger oscillations in the impaired system than in the original system, and as previously noted the impaired nephron 7 is exhibiting larger oscillations than nephron 8. Conversely, nephrons 3–6 exhibit a reduction in the oscillation
6.4 Impairments in an 8-nephron system

Figure 51: Variations in the WSFR of an eight-nephron system, in response to single impaired nephrons of different loop lengths.

Magnitude. Nephrons 1 and 2 have increased oscillations at the natural frequencies of loop lengths 1 and 3, but have decreased oscillations at their own natural frequency (loop length 0).

From the spikes at the natural frequency of loop length 3 that are present in the power spectral densities of other nephron pairs, it is evident that the infection of nephron 7 has affected the SNGFR of nephrons 1 and 2. Also, although nephrons 3–6 exhibit reduced oscillations in SNGFR, their spectral densities show no spikes at the natural frequency of the impaired nephron, which would appear to indicate that these nephrons are somehow less affected by the impaired nephron. The changes in spectral density peaks are summarised in Table 14.
Figure 52: Comparisons of WSFR between the original (unimpaired) eight-nephron system and the eight-nephron system where nephron 7 is impaired and $D_P = 0.5$.

<table>
<thead>
<tr>
<th>Nephron</th>
<th>Peak Increase</th>
<th>Nephron</th>
<th>Peak Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.25%</td>
<td>2</td>
<td>0.25%</td>
</tr>
<tr>
<td>3</td>
<td>-77%</td>
<td>4</td>
<td>-77%</td>
</tr>
<tr>
<td>5</td>
<td>-4.7%</td>
<td>6</td>
<td>-4.7%</td>
</tr>
<tr>
<td>7</td>
<td>183%</td>
<td>8</td>
<td>153%</td>
</tr>
</tbody>
</table>

Table 14: Increases in spectral density peaks in the SNGFR of nephrons in the system with one impaired nephron.
6.4 Impairments in an 8-Nephron System

(a) Nephrons 1 and 2 (loop length 0).  (b) Nephrons 3 and 4 (loop length 1).

(c) Nephrons 5 and 6 (loop length 2).  (d) Nephrons 7 and 8 (loop length 3).

Figure 53: The power spectral densities of SNGFR in the eight-nephron system with a single impaired nephron.

(a) Nephrons 1 and 2 (loop length 0).  (b) Nephrons 3 and 4 (loop length 1).

(c) Nephron 5 and 6 (loop length 2).  (d) Nephrons 7 and 8 (loop length 3).

Figure 54: The difference in power spectral densities of SNGFR between the original eight-nephron system and the system with one impaired nephron.

Returning attention to Fig. 52, despite changes in SNGFR spectral densities at several frequencies, the WSFR spectral density only exhibits an increase at the natural frequency of the impaired nephron, and a decrease in the average WSFR of 0.7%. Despite the spectral
densities of individual nephrons varying from -77% to 183%, the average WSFR is not significantly affected, demonstrating that this eight-nephron system can exhibit stable whole-system behaviour when the function of an individual nephron is impaired.

6.4.2 Two impaired nephrons

The second experiment was to determine the effects of two impaired nephrons on the same system. In this experiment, $D_P$ was kept fixed at 0.5 and for each pair of nephrons in the system, a simulation was run where both nephrons were impaired at $t = 1,200$ and the simulation ran until $t = 3,000$.

A comparison of the WSFR for each of these simulations is presented in Fig. 55—Fig. 55a shows the average WSFRs and Fig. 55b shows the magnitude of the oscillations in WSFRs. From Fig. 55a it is clear that the impairment decreased the average WSFR in proportion to the lengths of the infected nephrons’ loops of Henle, as was the case for a single impaired nephron. The magnitude of WSFR oscillations (Fig. 55b) also shows a similarity to the previous experiment, as it was not affected by the impairment unless at least one of the longest-loop nephrons was impaired.

A comparison between the original system and the impaired system where both longest-loop nephrons are impaired will now be given. This particular impaired system was chosen as it exhibits the lowest average WSFR of the systems under consideration (see Fig. 55a).

The WSFR of the two systems are shown in Fig. 56a, where the decrease in average WSFR due to the impairment is apparent. The power spectral densities of the two WSFRs are shown in Fig. 56b, and the impaired system can be seen to have peak seven times larger at the natural frequency of loop length 3. This peak is approximately five times larger than the equivalent peak in the system with a single impaired nephron (Fig. 56b), indicating that the two impaired nephrons have a much larger influence on the system dynamics than the single impaired nephron in Sect. 6.4.1.

The analysis now turns to the SNGFRs of the individual nephrons. The power spectral densities of the SNGFRs are presented in Fig. 57. This figure shows that while nephrons 3–8 are essentially oscillating at a single frequency, nephrons 1 and 2 (the nephrons with the shortest loops) are oscillating at a combination of different frequencies. These spectral densities exhibit three main differences from the previous system that contained only a single impaired nephron:

- Nephrons 1 and 2 feature a significantly larger peak at the natural frequency of loop length 3 (compare Fig. 53a and Fig. 57a).
- Nephrons 3 and 4 exhibit smaller peaks, with differing magnitudes, which accentuate the small peaks at the natural frequencies of loop lengths 0 and 3 (compare Fig. 53b and Fig. 57b). These additional peaks are present in the system with one impaired nephron (Fig. 53b), but were minuscule in comparison to the dominant peak.
6.4 Impairments in An 8-Nephron System

(a) The average WSFR over the simulations.

(b) The size of oscillations in WSFR over the simulations.

Figure 55: Variations in the WSFR of an eight-nephron system, in response to pairs of impaired nephrons of different loop lengths.

- Nephrons 7 and 8 again exhibit peaks of different magnitude (compare Fig. 53d and Fig. 57d), but in this system both nephrons have identical impair states.

The difference in spectral densities between nephrons 3 and 4, and between nephrons 7 and 8 is unexpected—both pairs of nephrons have identical impair states—but one possible explanation is that the different peak sizes may be due to the relationship between the phase of the individual nephron and the phases of the other nephrons in the system. This phenomena will not be investigated in any further detail here—the changes in spectral densities will now be examined.

Plots of the differences in spectral densities between the original and impaired systems are given in Fig. 58. The plot for nephrons 1 and 2 exhibit the large peak at the natural frequency of loop length 3.
(a) The WSFR in both the original and impaired eight-nephron systems.

(b) The power spectral density of WSFR in both the original and impaired eight-nephron systems.

Figure 56: Comparisons of WSFR between the original eight-nephron system and the eight-nephron system where both longest-looped nephrons are impaired and $D_P = 0.5$.

(see Fig. 58a) and also show that the increase at their own natural frequency is approximately half of the increase shown when a single nephron is impaired (compare Fig. 54a and Fig. 58a). Nephrons 3–6 again exhibit a decrease at their natural frequencies (Fig. 58b, Fig. 58c) and there is a difference in the magnitude of the decreases in nephrons 3 and 4, but it is small enough to be almost invisible (Fig. 58b). The plot for nephrons 7 and 8 clearly shows the difference in peak amplitudes (Fig. 58d) that was observed in Fig. 57d.

It is again evident that nephrons 7 and 8 have larger oscillations in the impaired system than in the original system, and the impairment of nephrons 7 and 8 has affected the SNGFR of nephrons 1 and 2. Although nephrons 3 and 4 do not exhibit an increase at the natural
6.4 Impairments in an 8-Nephron System

Figure 57: The power spectral densities of $\text{SNGFR}$ in the eight-nephron system where both longest-loop nephrons are impaired.

Figure 58: The difference in power spectral densities of $\text{SNGFR}$ between the original eight-nephron system and the system with two impaired nephrons.
Table 15: Increases in spectral density peaks in the SNGFR of nephrons in the system with two impaired nephrons.

<table>
<thead>
<tr>
<th>Nephron</th>
<th>Peak Increase</th>
<th>Nephron</th>
<th>Peak Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-3%</td>
<td>2</td>
<td>-3%</td>
</tr>
<tr>
<td>3</td>
<td>-95%</td>
<td>4</td>
<td>-95%</td>
</tr>
<tr>
<td>5</td>
<td>-10%</td>
<td>6</td>
<td>-10%</td>
</tr>
<tr>
<td>7</td>
<td>140%</td>
<td>8</td>
<td>111%</td>
</tr>
</tbody>
</table>

Returning attention to Fig. 56, despite changes in SNGFR spectral densities at several frequencies, the WSFR spectral density only exhibits an increase at the natural frequency of the impaired nephrons, and a decrease in the average WSFR of 1.4%—twice the decrease that arose with a single impaired nephron, even though the WSFR spectral density showed a peak five times larger than in the system with a single impaired nephron (compare Fig. 52b and Fig. 56b). Despite the spectral densities of individual nephrons varying from -95% to 140%, the average WSFR is not significantly affected, demonstrating that this eight-nephron system exhibits stable whole-system behaviour when the function of the two longest-loop nephrons are impaired.

6.4.3 Summary of experiments

The results from both experiments indicate that the dynamics of this eight-nephron system are more sensitive to changes in the longer-looped nephrons than the shorter-looped nephrons. A possible explanation for this observation is that the longer-looped nephrons play a more dominant role in the maintenance of the salt gradient in the medullary fluid, and that any change to their behaviour will therefore have a larger impact on the system dynamics. The SNGFR of the longer-looped nephrons also oscillates more slowly than the shorter-looped nephrons—due to the longer time delay between the filtrate entering the tubule and reaching the macula densa—and accordingly an impaired nephron might only affect neighbouring nephrons with shorter loops of Henle, as the shorter-looped nephrons can respond to imbalances more quickly than the longer-looped nephrons.

Similarly, although the average WSFR only decreased by 0.7–1.4%, the amplitude of the oscillations in WSFR increased, which may indicate that the system as a whole was producing a larger dynamic response to the changed conditions. This might mean that the impaired systems are less able to respond to further changes in conditions, as they are oscillating closer to their maximal range. Such an effect of impairments on multi-nephron systems—that the system may be less able to respond to changes in conditions, rather than just exhibiting changed steady-state dynamics—will not be investigated further here.
6.5 IMPAIRMENTS ON A LARGER SCALE

The second investigation was to analyse the effect that impaired nephrons have at a larger scale—the model used for this experiment was a 72-nephron system, consisting of nine columns arranged in a $3 \times 3$ grid, the structure of each column being identical to that of the eight-nephron system used in Sect. 6.4 (see Fig. 50). The $2D$ arrangement of the interstitial fluid columns is shown in Fig. 59a and the arterial tree structure is shown in Fig. 59b. In each simulation, nephrons were impaired at $t = 1,200$ with $D_P = 0.5$, and the simulations were run until $t = 3,000$.

![Figure 59](image-url)

(a) The $2D$ arrangement of interstitial fluid columns.  
(b) The arterial tree connecting the columns.

Figure 59: The high-level structure of the 72-nephron model. Each column is assigned a number 1–9.

Two separate experiments were conducted. The first investigated the effects of impairing all nephrons in the model whose loops of Henle were the same length, to determine how significantly the length of the loop affects the system dynamics. The difference between this experiment and the one conducted on the eight-nephron system (Sect. 6.4.1) is that the 72-nephron model incorporates inter-column diffusion and a more complex arterial tree, which may reduce the effect of the impairment.

The second experiment investigated the effects of impairing the nephrons in an entire column, to determine how the impairment affects neighbouring columns. This experiment allowed the effects of the $2D$ column arrangement and the arterial tree structure to be compared. These experiments are now presented and compared, and the effects of impaired nephrons on the 72-nephron system are then summarised.
6.5.1 Impaired nephron pairs

In this experiment, for each loop length \( L \) (see Fig. 50) all of the nephrons whose loops were length \( L \) were impaired at \( t = 1, 200 \). This experiment is essentially a larger-scale equivalent of the experiment detailed in Sect. 6.4.2.

A comparison of the WSFR for each of these simulations is presented in Fig. 60. The average WSFR decreased in proportion to the loop length of the impaired nephrons (Fig. 60a), as per the experiments in Sect. 6.4. When the longest-loop nephrons in the model were impaired, the decrease in WSFR was 1.3%, which is almost identical to the decrease observed in Sect. 6.4.2. As shown in Fig. 60b, the size of the oscillations in WSFR varied with no discernible regularity, in contrast to the eight-nephron system (Fig. 55b).

The remainder of this analysis contrasts the dynamics of the original 72-nephron system with the dynamics of the system where all nephrons of loop length 3 are impaired, as this system demonstrated the largest decrease in WSFR (see Fig. 60).

The WSFR of both the original and impaired systems are presented in Fig. 61, where the decrease in average WSFR is not immediately apparent (Fig. 61a). The power spectral densities of the two WSFRs (Fig. 61b) show that the impaired nephrons have caused larger peaks at the natural frequencies of loop lengths 1 and 3. There is also an additional minor spike at a normalised frequency of approximately 0.033, which is a higher frequency than even that of the shortest-loop nephrons. The increases shown here are much smaller than the increased spikes that arose when the two longest-loop nephrons were impaired in the eight-nephron model (compare Fig. 56b and Fig. 61b).

The analysis now turns to the WCFRs of the individual columns in the model, which is one level of hierarchy lower than the WSFR but is also a level of hierarchy above the SNGFR. At this level of hierarchy the WCFR is directly comparable to the WSFR of the eight-nephron models that were investigated in Sect. 6.4, and thus any observations made in these previous experiments may be relevant here.

The spectral densities of the WCFR for each column in the model are shown in Fig. 62. Columns 1–3 exhibited two dominant peaks, both of which have increased in size in the impaired system—the increase in the peak at the natural frequency of the nephrons with loop length 3 has increased more than the secondary peak at the natural frequency of nephrons with loop length 1 (Fig. 62a). Columns 4–9 all exhibited three dominant peaks—at the natural frequencies of nephrons with loop length 0, 1 and 2. There was a difference in relative sizes of the peaks, with columns 4–6 having a larger peak at the frequency of loop length 1 (Fig. 62b), and columns 7–9 having a larger peak at the frequency of loop length 0 (Fig. 62c).

All of the columns exhibited a larger spike at the natural frequency of the longest-loop nephrons (Fig. 63), which is consistent with the experiments conducted on the eight-nephron system. As was shown in Sect. 6.4, this does not mean that nephrons with different natural frequencies are unaffected. Apart from this trend, there is no observable common pattern in the changes in power spectral densities.
Figure 60: Variations in the WSFR in the 72-nephron system where all nephrons of a specific loop length are impaired.

It is also important to note that the lowest-frequency peak in all three graphs covers the natural frequencies of nephrons with loop lengths 2 and 3. For columns 1–3, the loop length 3 frequency dominated and the loop length 2 frequency formed a shoulder (Fig. 62a), while for columns 4–9 it was the loop length 2 frequency that dominated and the loop length 3 frequency that formed a shoulder (Fig. 62b and Fig. 62c). This indicates that the impaired nephrons are possibly having a larger effect on columns 1–3, as these were the only columns to show a clear increase in the peak size for the loop length 3 frequency.

However, as was demonstrated in earlier experiments (Sect. 6.4), an increased peak at the loop length 3 frequency does not mean that the impaired nephrons are having a significant effect—in Fig. 56b
EMERGENT EFFECTS OF LOCALISED IMPAIRMENT

Figure 61: Comparison of the WSFR in the original 72-nephron system to that of the 72-nephron system where all nephrons of a specific loop length are impaired.

(a) The WSFR in both the original and impaired 72-nephron systems.

(b) The power spectral densities of WSFR in both the original and impaired 72-nephron systems.

In summary, the WSFR and WCFR power spectral densities showed smaller differences between the original and impaired 72-nephron systems than those observed in the equivalent experiment on the eight-nephron system (Sect. 6.4.2), and the decrease in average WSFR
Figure 62: A comparison of the power spectral densities of WCFR between the original 72-nephron system and the system where all nephrons of a specific loop length are impaired.
Figure 63: The difference in power spectral densities of WCFR between the original 72-nephron system and the system where all nephrons of a specific loop length are impaired.
6.5 IMPAIRMENTS ON A LARGER SCALE

6.5.2 Impaired columns

In this experiment, for each column in the model a simulation was performed where all of the nephrons in that column were impaired at \( t = 1, 200 \). A comparison of the WSFR for each of these simulations is presented in Fig. 65. Fig. 65a shows that the average WSFR decreased the most when the bottom-most columns in the arterial tree (columns 7–9) were impaired—a maximal decrease of 0.3%. All of the impaired simulations exhibited a smaller range in WSFR than the original system (Fig. 65b), with the smallest range being produced...
when one of columns 4–6 was impaired. A potential explanation for this observation is that columns 4–6 undergo inter-column diffusion with columns at both top and bottom branches of the arterial tree, possibly resulting in a more wide-spread influence over the dynamics of the other columns.

![Graph](image)

(a) The average WSFR over the simulations.

![Graph](image)

(b) The size of oscillations in WSFR over the simulations.

Figure 65: Variations in the WSFR in the 72-nephron simulations where all of the nephrons in a particular column \( (x, y) \) are impaired.

The remainder of the analysis will contrast the dynamics of the original 72-nephron system with the dynamics of the system where all nephrons in column 9 are impaired, as this is one of the three systems that demonstrates the largest decrease in WSFR (shown in Fig. 65a).

The WSFRs of the original and impaired systems are plotted in Fig. 66a—the decrease in average WSFR is not discernible, as the difference is 0.3%. The power spectral densities of the two WSFRs are
shown in Fig. 66b, where the impaired system is again shown to have
an enlarged peak at the natural frequency of loop length 3, and also
a decreased peak at the natural frequency of loop length 1—a trend
not observed in the previous experiments.

Figure 66: Comparison of the WSFR in the original 72-nephron system
and the system where all of the nephrons in column 9 are impaired.

The analysis now turns to the WCFRs of the individual columns in the
model. The power spectral densities of the WCFRs—shown in Fig. 67—
were very similar to those in the previous experiment (Fig. 62), except
that the mountain at the natural frequencies of loop lengths 2 and
3 exhibited a small valley between the two peaks. Apart from these
changes, there appears to be little difference between the spectral
densities of the WCFRs in the original and impaired systems.

Fig. 68 shows the changes in WCFR spectral densities as a result of
the impairment in column 9. In comparison to the changes in WCFR
Figure 67: A comparison of the spectral densities of WCFR between the original 72-nephron system and the system where all of the nephrons in column 9 are impaired.
Figure 68: The difference in power spectral densities of WCFR between the original 72-nephron system and the system where all of the nephrons in column 9 are impaired.
Figure 69: The difference in the power spectral density of column 9 in the original 72-nephron system and the system where all of the nephrons in column 9 are impaired.

spectral densities from the previous experiment (Fig. 63), several observations can be made.

- Columns 1–3 exhibit similar trends in both experiments (compare Fig. 68a and Fig. 63a), but the magnitude of the increases and decreases in peak size are approximately 5–10 times smaller.

- Columns 4–6 also exhibit similar trends in both experiments, but again the magnitude of the changes is reduced—this time 2–3 times smaller—and there is no peak at the natural frequency of loop length 0 in this experiment.

- Columns 7–8 do not exhibit trends similar to those in the previous experiment, and are also the only columns that exhibit
larger changes in these spectral densities than in the previous experiment. This is most likely due to the impaired state of column 9 exerting a larger influence on these two columns—via the arterial tree—than the effects of the impairment in the previous experiment.

The spectral density of the WCFR of column 9 is shown in Fig. 69a, and while it shares a similar shape with the WCFRs of the other columns (Fig. 67), it is the only column that exhibited such marked increases in the natural frequencies of loop lengths 2 and 3. The changes in the spectral density are shown in Fig. 68, which is unique in that all peaks except one are positive and that the peak size is inversely proportional to the normalised frequency.
A direct comparison of the individual WCFRs in the original and impaired systems is given in Fig. 70. Only column 9 shows evidence of a change in WCFR (Fig. 70a), which would indicate that the other columns are unaffected by the impairment of column 9. However, examining the size of the oscillations in WCFR reveals that column 9 has influenced other columns (Fig. 70b)—columns 1–3 show a decrease of 0.5%, columns 4–6 show a decrease of 2.6%, column 7 shows an increase of 0.6%, column 8 an increase of 0.8% and column 9 an increase of 12%.

Ignoring column 9, the changes in WCFR oscillation size are essentially uniform for all of the columns sharing the same branch of the arterial tree—there is only a 0.03% difference in the oscillation sizes of columns 5 and 6, even though column 6 undergoes inter-column diffusion with the impaired column and column 5 does not. This would appear to indicate that the inter-column diffusion mechanism is able to dampen out local changes, and that the arterial tree has a more dominant role in propagating the effects of the impairment across the system.

6.5.3 Summary of experiments

The first experiment conducted on the 72-nephron system—the longest-looped nephrons in each column were impaired—demonstrated a 1.3% decrease in average WSFR. The same experiment, when conducted on the eight-nephron system (Sect. 6.4.2), demonstrated a 1.4% decrease in average WSFR. These values were expected to be similar, as the individual columns in this model share a structure identical to that of the eight-nephron model, and the impairments in both experiments were identical. The inter-column diffusion could not have caused any difference in average WSFR, since the interstitial fluid in each column evolved identically over time, due to the identical structure and impairment shared by all of the columns. It is possible that the difference in average WSFR between these experiments is due to the arterial tree structure connecting the columns, but the difference is so small that a more detailed investigation is required to determine this for certain.

In contrast, when the impairment was localised to a single column in the 72-nephron system, the impairment did not appreciably affect the dynamics of the system—average WSFR was decreased by only 0.3% and neighbouring columns were not significantly affected (see Fig. 70). This is due to the inter-column diffusion (depicted in Fig. 16), which acts as an averaging mechanism for solute concentration gradients in the interstitial fluid. The arterial tree was able to affect the magnitude of the oscillations in WSFR (shown in Fig. 65b), but the inter-column diffusion was able to keep the average WSFR almost constant (shown in Fig. 65a), regardless of the choice of impaired column. As the choice of column affects the number of neighbouring columns with which inter-column diffusion occurs, this indicates that the inter-column diffusion is a relatively strong agent for overcoming the localised effects of this impairment.
As mentioned at the start of the chapter, the impairment presented here and the accompanying experiments serve only to illustrate the approach of modelling and analysing the effects of a localised impairment on multi-nephron systems. However, the experiments conducted here demonstrated that a system consisting of a single column of interstitial is strongly resistant to the effects of the impairment. Furthermore, for a system consisting of multiple columns, the inter-column diffusion mechanism further reduced the effects of the impairment, to the point that there were little discernible effects on the whole-system dynamics.

Without making any predictions or observations concerning the real-world effects of renal diseases on the kidney, these experiments demonstrated that the multi-nephron systems maintain remarkably stable dynamics when the function of individual nephrons are impaired. Despite the simplicity of the impairment model that was used, we contend that this stability is a feature of the model—and of the kidney proper—rather than a deficiency in the impairment model.
7.1 INTRODUCTION

Since one of the motivations for this model is to make whole-kidney simulations computationally tractable, our attention now turns to how the performance of the simulations scale as the size of the model increases. The experiments conducted in Chapter 5 and Chapter 6 have only examined the behaviour of the model in the context of multi-nephron systems whose sizes pale in comparison to that of an entire kidney. To discuss the simulation performance, several distinctions must be made:

**MODEL** The network model, as presented in Chapter 4.

**ALGORITHM** A computational method for calculating the dynamics of a given model.

**IMPLEMENTATION** A computer program that executes an algorithm.

**PROCESS** A sequence of computations, performed in serial.

**SIMULATION** The act of using an implementation to calculate the dynamics of a given model.

This chapter begins by presenting the algorithm that was used for the experiments in Chapter 5 and Chapter 6, followed by a performance analysis of the resulting implementation. Despite the fact that this implementation is single-threaded, and is therefore unable to make use of more than one central processing unit (CPU), it is shown that using this implementation to simulate a whole-kidney model is computationally tractable.

Secondly, the simulation performance can be greatly improved by utilising more than one CPU, and a new algorithm is presented that performs each time-step as a number of concurrent processes, each of which can be performed on a different CPU. This algorithm was implemented using the Join Calculus [60, 59], which allows a single simulation to be distributed over multiple computers. A performance analysis demonstrates that this results in a significant performance increase when more than one CPU is available.

Finally, to calculate the dynamics of an entire kidney, it is not sufficient for the simulation to be computationally tractable—the model must first be constructed. The structure of the models used in Chapter 5 and Chapter 6 were manually specified—the spatial parameters of each tubule segment, each nephron and the arterial tree—an approach which clearly will not scale to models of $10^6$ nephrons. Although no solution has currently been implemented, two techniques for automatically generating such large models are proposed.
7.2 PERFORMANCE OF THE IMPLEMENTATION

7.2.1 The single-threaded algorithm

A pseudocode description of the single-threaded algorithm is given in Listing 7.1.

```python
# determine the resistance of the arterial tree
for column in model.columns:
    for nephron in column.nephrons:
        # calculated by an update rule (Eq. 4.5)
        nephron.update_resistance()
        # calculated from the nephron resistances (Fig. 14)
        # Ohm’s law and Kirchoff’s laws are used
        column.update_resistance()
        # calculated from the column resistances (Fig. 15)
        # Ohm’s law and Kirchoff’s laws are used
        RT = model.update_resistance()

        # update the model parameters (Sect. 4.4.3)
        model.update_params()

        # generate the input for the next time-step
        # volume = \( \frac{P_A - P_V}{RT} \) (Sect. 4.2.2)
        input = model.input_blood(RT)

        # distribute the blood to the nephrons
        for column in model.columns:
            # depends on the tree of column resistances (Fig. 15)
            column_input = model.distribute(input, column)
            for nephron in column.nephrons:
                # depends on the tree of nephron resistances (Fig. 14)
                nephron_input = column.distribute(column_input, nephron)
                nephron.set_input(nephron_input)

        # simulate the time-step for each tubule segment (Sect. 4.4.5)
        # also update the state variables (Sect. 4.4.4)
        for column in model.columns:
            column.simulate_timestep()

        # perform diffusion between neighbouring columns (Fig. 16)
        for column1, column2 in model.column_neighbours:
            column1.diffuse_with(column2)

        # update the nephron disease states (Sect. 6.1)
        for column in model.columns:
            for nephron in column.nephrons:
                nephron.update_disease_state()
```

Listing 7.1: The single-threaded algorithm in pseudocode.
7.2.2 Performance analysis

The single-threaded algorithm was implemented in OCaml [101] and was compiled with the high-performance native-code compiler OCAMLOPT. The simulations were performed on an Intel Core 2 Duo E6600 with 2Gb of RAM, running Ubuntu Linux. The parameter values do not affect the simulation performance, as they do not affect the calculations used to solve the model equations.

The memory usage of the simulations (shown in Fig. 71a) grows linearly with the number of nephrons in the model, and the number of columns into which the model is divided appears to have little effect. However, the execution times show significant differences, depending on the number of columns in the model (Fig. 71b). The execution time grows at $O(n^3)$ for $n$ nephrons per column, as demonstrated by the curves fitted to the execution time plots (see Fig. 72). This is to be expected, as solving the $n \times n$ matrix governing the loops of Henle (described in Sect. 4.4.5) is solved by Gaussian elimination using the standard algorithm, which has a complexity of $O(n^3)$.

As demonstrated by Fig. 71c, when the number of nephrons in each column is held constant, the execution time grows linearly with the number of columns in the model. Thus, for a model containing $c$ columns and $n$ nephrons per column, the execution time for the simulation is $O(cn^3)$. This explains why the $n^3$ coefficients in Fig. 72 decrease by a factor of four when $c$ is doubled—$n$ is halved to conserve the number of nephrons in the model, and so:

\begin{align*}
    c' &= 2 \times c \\
    n' &= \frac{n}{2} \\
    O\left((c'n^3)^3\right) &= O\left(2c \times \left(\frac{n}{2}\right)^3\right) = O\left(\frac{cn^3}{4}\right)
\end{align*}

For example, consider two models of 100,000 nephrons (Table 16). The first consists of 100 columns arranged in a $10 \times 10$ grid with 1,000 nephrons per column; the second consists of 10,000 columns arranged in a $100 \times 100$ grid with 10 nephrons per column. As the performance of a model is $O(cn^3)$, the performance of the first model is $O(10^{11})$, compared to $O(10^7)$ for the second model. This is not only a huge improvement in performance, as the structure of the second model more closely approximates a continuous body of interstitial fluid—the performance of the model improves the more the interstitial fluid structure approximates a continuous model.

Using this implementation, a simulation of a 512-nephron model (8 nephrons per column) progressed at 444 nephron time-steps per second and used 28Mb of RAM (56kb per nephron). Assuming that the linear relationships in Fig. 71a and Fig. 71c hold for a model of $10^6$ nephrons, this indicates that the simulation of a single time-step for a whole-kidney model would require 37.5 minutes and 53.4Gb of RAM. These orders of magnitude demonstrate that the implementation scales up to a whole-kidney model. We are not aware of any existing nephron models for which a whole-kidney simulation is computationally feasible.
Figure 71: Comparing 1-column, 2-column and 4-column simulations.
Figure 72: Execution time is $O(n^3)$ for $n$ nephrons per column.
Table 16: Performance comparison of two models of identical size.

<table>
<thead>
<tr>
<th></th>
<th>Model 1</th>
<th>Model 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nephrons</td>
<td>10^5</td>
<td>10^5</td>
</tr>
<tr>
<td>c</td>
<td>10^2</td>
<td>10^4</td>
</tr>
<tr>
<td>n</td>
<td>10^3</td>
<td>10^1</td>
</tr>
<tr>
<td>Time</td>
<td>O(10^{11})</td>
<td>O(10^7)</td>
</tr>
</tbody>
</table>

7.3 IMPROVING PERFORMANCE WITH CONCURRENCY

Although whole-kidney simulations are achievable with the monolithic algorithm, the performance of such simulations can readily be improved. While reductions in execution time may be achieved by optimising the monolithic implementation, we show that much larger improvements can be achieved by dividing the simulation into multiple processes that can be executed concurrently. If the processes can be distributed over multiple computers—employing significantly more resources than available on a single computer—the resulting performance gain of the distributed system can be remarkable [94].

Typical distributed systems such as Berkeley Open Infrastructure for Network Computing (BOINC) [35, 7], SETI@home [94, 8] and KidneyGrid [42, 43], and distributed algorithms such as MapReduce [51] operate by dividing the computational work into a number of distinct processes, executing each process independently, and performing some analysis or calculation on the results.

In contrast, the model presented in Chapter 4 requires the model state at time $t$ to be calculated before the state at time $t + \Delta$ can begin to be calculated. As a result, if the simulation were broken into multiple processes, the processes would need to communicate with each other after every time-step. This is in contrast to the distributed systems described previously, which perform multiple jobs concurrently and then combine the results in some manner. Accordingly, a concurrent algorithm for the model simulation will necessarily differ from such systems and cannot be implemented using an identical approach.

7.3.1 The concurrent algorithm

The single-threaded algorithm Listing 7.1 can be divided into a number of separate processes, based on the hierarchical structure of the model. Once the input has been generated and distributed to the columns, the time-step can be simulated for each column independently, with the exception of the inter-column diffusion. The inter-column diffusions can also be performed independently: for each pair of neighbouring columns, assign one the “master”, which is responsible for simulating the diffusion with the “slave” (see Fig. 73). This algorithm consists of a process for each column in the model (referred to as column processes and abbreviated to “Col1”, “Col2”, etc., in diagrams), and a process for the generation and distribution
of the model input (referred to as the “controller” and abbreviated to “Ctrl” in diagrams). The algorithm is presented in pseudocode form in Listing 7.2 and Listing 7.3, and the communication between the processes is shown in Fig. 74 and Fig. 75. To reduce the delay between time-steps, the column processes can be changed so that once they receive input, they calculate the resistance of the arterial tree and immediately send this value to the controller (demonstrated in Fig. 76).

```
# receive the resistances of each column in the model
for column in model.columns:
    resistance = receive_message(column, send.resistance)
    R_T = update.resistance(column, resistance)

# generate the input for the next time-step
# volume = \( \frac{P_A - P_V}{R_T} \) (Sect. 4.2.2)
input = model.input.blood(R_T)

# distribute the blood to the columns
for column in model.columns:
    column_input = model.distribute(input, column)
    send_message(column, column_input)
```

Listing 7.2: The controller process in pseudocode.

```
# receive the resistances of each column in the model
for column in model.columns:
    resistance = receive_message(column, send.resistance)
    R_T = update.resistance(column, resistance)

# generate the input for the next time-step
# volume = \( \frac{P_A - P_V}{R_T} \) (Sect. 4.2.2)
input = model.input.blood(R_T)

# distribute the blood to the columns
for column in model.columns:
    column_input = model.distribute(input, column)
    send_message(column, column_input)
```

The presentation of the column algorithm (Listing 7.2) has ignored the complexities of performing the inter-column diffusion. Each column must not only perform the diffusion between itself and neighbouring slaves, it must also communicate with neighbouring masters (see Fig. 75). This more detailed algorithm is presented in Listing 7.4.

A significant feature of this concurrent algorithm is the total absence of shared state between processes—the only shared data is passed between processes using messages. This greatly simplifies
# determine the resistance of the arterial tree
for nephron in column.nephrons:
    # calculated by an update rule (Eq. 4.5)
    nephron.update_resistance()
    # calculated from the nephron resistances (Fig. 14)
    \( R_C = \text{column.update.resistance()} \)

# send the column resistance to the controller
send_message(controller, \( R_C \))

# receive the input blood for the next time-step
receive_message(column_input)

# distribute the blood to the nephrons
for nephron in column.nephrons:
    # depends on the tree of nephron resistances (Fig. 14)
    nephron_input = column.distribute(column_input, nephron)
    nephron.set_input(nephron_input)

# simulate the time-step for each tubule segment (Sect. 4.4.5)
# also update the state variables (Sect. 4.4.4)
column.simulate_timestep()

# perform diffusion between neighbouring columns (Fig. 16)
for neighbour in column.slaves:
    column.diffuse_with(slave)

# update the nephron disease states (Sect. 6.1)
for nephron in column.nephrons:
    nephron.update_disease_state()
7.3 Improving Performance with Concurrency

![Diagram of communication between columns](image1)

**Figure 75:** Communication between columns when performing the inter-column diffusion.

![Diagram of communication reordering](image2)

**Figure 76:** Reordering the communication in Fig. 74, in order to reduce the delay between time-steps.

Associated with concurrent programs. Perhaps the biggest advantage is that the implementation can easily be distributed across multiple computers, since the lack of shared memory between computers has no impact.

### 7.3.2 Communication in the Concurrent Algorithm

The concurrent algorithm that has been described involves regular communication between the individual processes—for each time-step, two messages are sent between each column and the controller (Fig. 74) and three messages are sent between each pair of neighbouring columns (Fig. 75). If the columns are distributed across a network of computers, the limitations of the network hardware can impact upon the performance of this communication, and thus reduce the simulation performance.
Listing 7.4: The algorithm for inter-column diffusion in pseudocode.

```python
# save the current state of the interstitial fluid
F_C = copy(column.interstitial_fluid)

# ask neighbouring slaves for their interstitial fluid
for neighbour in column.slaves:
    send_message(neighbour, request_interstitial_fluid)

# count how many slaves and masters have not completed the diffusion
slaves = column.slave_count
masters = column.master_count

while true:
    # receive messages from neighbouring columns
    message = receive_message()

    # a master has requested the column's interstitial fluid
    if message == request_interstitial_fluid:
        send_message(message.sender, F_C)
    # a master has adjusted the interstitial fluid
    elif message == adjust_interstitial_fluid:
        column.adjust_interstitial_fluid(message.changes)
        master_count -= 1
    # a slave has sent its interstitial fluid
    elif message == send_renal_fluid:
        local, remote = diffusion_changes(F_C, message.fluid)
        # inform the slave to adjust its interstitial fluid
        send_message(message.sender, adjust_renal_fluid(remote))
        # adjust this column's interstitial fluid
        column.adjust_interstitial_fluid(local)
        slave_count -= 1
    # stop when all diffusions have been completed
    if slaves == 0 and masters == 0:
        break
```

Table 17: Message content in inter-process communication. The model supports $S$ solutes and each column has $L$ layers of interstitial fluid.

<table>
<thead>
<tr>
<th>Message</th>
<th>Data</th>
<th>Size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send resistance</td>
<td>double</td>
<td>8</td>
</tr>
<tr>
<td>Send input</td>
<td>double list (fluid)</td>
<td>$8 \times S$</td>
</tr>
<tr>
<td>Request interstitial fluid</td>
<td>none</td>
<td>0</td>
</tr>
<tr>
<td>Send interstitial fluid</td>
<td>fluid list</td>
<td>$8 \times S \times L$</td>
</tr>
<tr>
<td>Adjust interstitial fluid</td>
<td>fluid list</td>
<td>$8 \times S \times L$</td>
</tr>
</tbody>
</table>

The content of each message sent between the individual processes is documented in Table 17. Based on this data, the maximum volume of data sent by a column in a single time-step is given by Eq. 7.4.

$$32 \times S \times L + 8 \times S + 8$$  \hspace{1cm} (7.4)
The model currently supports eight solutes (Sect. 4.4.3) and the models used in Chapter 5 and Chapter 6 used up to sixteen layers of interstitial fluid, resulting in a maximum volume of 4168 bytes (4.07 kb). To examine the worst-case volume of communication, consider a whole-kidney model (i.e., $10^6$ nephrons) that consists of $10^6$ columns, each containing a single nephron. Using Eq. 7.4, the volume of data sent at each time-step would be 3.88Gb—such a large volume of communication would certainly reduce the simulation performance.

This volume can be decreased by reducing the number of columns in the model, but there are two further options that do not affect the model structure: locating multiple columns on the same computer (i.e., sharing a single CPU or using multiple CPUs); and allowing columns to simulate the behaviour of multiple columns. Both of these options were explored with the implementation.

### 7.3.3 Performance of the implementation

Although the concurrent algorithm differs significantly from the original algorithm (Listing 7.1), the concurrent implementation shares most of its code with the original implementation (Sect. 7.2.2). The concurrent implementation was written in JoCaml [110, 147, 100], an extension of OCaml that incorporates the Join Calculus [60, 59] to allow for concurrent and distributed programming.

The Join Calculus—like other process calculi such as CCS [120], the $\pi$ Calculus [121, 122], the Fusion Calculus [140, 141] and the Ambient Calculus [34]—is used to model concurrent systems. The Join Calculus is equivalent to the $\pi$ Calculus [59], but is easier to implement in a distributed setting [59].

The Join Calculus uses a message-passing concurrency model very similar to the one used to present the concurrent algorithm (Listing 7.2 and Listing 7.3), which greatly simplified the implementation process. To illustrate how well the JoCaml semantics match the concurrent algorithm, the implementation of the inter-column diffusion (Listing 7.4) is presented in Listing 7.5. The only difference is that the loop and the variables \textit{slaves} and \textit{masters} are captured by the message \textit{diff.state}(column, $F_C$, slaves, masters).

Amdahl’s law [6] establishes the theoretical maximum speedup that can be gained by performing a computation with multiple processors, where a fraction $p$ of the computation can be executed in parallel and the computation is performed on $N$ identical processors (Eq. 7.5). According to this law, we expect to see a significant speedup as the simulation is distributed over a greater number of processors, although the return diminishes and approaches a limit (see Eq. 7.6 and Fig. 77).

\[
S(N) = \frac{1}{(1 - p) + p/N}
\]  
\[
\lim_{N \to \infty} S(N) = \frac{1}{1 - p}
\]  

This law is based on the implicit assumption that $p$ is independent of $N$, which is only the case if the size of the problem is fixed. If this is not the case and the problem size can scale with the number of
# start the inter-column diffusion
diff.start(column) =
  # save the current state of the interstitial fluid
  F_C = copy(column.interstitial_fluid)
  # ask neighbouring slaves for their interstitial fluid
  for neighbour in column.slaves:
    spawn request_fluid(column, neighbour)
  # wait to receive messages from neighbouring columns
  diff.state(column, F_C, column.slaves, column.masters)

  # no more masters or slaves to wait for
  diff.state(column, F_C, 0, 0) =
    # the inter-column diffusion is complete
    diff.complete(column)

  # a master requests the column's interstitial fluid
  diff.state(column, F_C, slaves, masters)
  & request_fluid(master, column) =
    # send the interstitial fluid to the master
    send_fluid(master, column, F_C)
    # still have to wait for all of the masters and slaves
    diff.state(column, F_C, slaves, masters)

  # a master changes the column's interstitial fluid
  diff.state(column, F_C, slaves, masters)
  & adjust_fluid(master, column, changes) =
    # adjust the interstitial fluid
    column.adjust_fluid(changes)
    # there is now one less master to wait for
    diff.state(column, F_C, slaves, masters - 1)

  # a slave sends its interstitial fluid
  diff.state(column, F_C, slaves, masters)
  & send_fluid(column, slave, fluid) =
    # calculate the changes to the fluid in both columns
    local, remote = diffusion_changes(F_C, message.fluid)
    # inform the slave to adjust its interstitial fluid
    adjust_fluid(column, slave, remote)
    # adjust this column's interstitial fluid
    column.adjust_interstitial_fluid(local)
    # there is now one less slave to wait for
    diff.state(column, F_C, slaves - 1, masters)

Listing 7.5: The JoCaml implementation of inter-column diffusion in pseudocode form.

processors (e.g., the complexity of the model may be increased by refining the equations, incorporating additional equations, or augmenting the structure), this will usually increase the relative amount of the computation that can be executed in parallel. As a result, the speedup gained by using additional processors can significantly exceed the original limit proposed by Amdahl’s law (e.g., by an order of magnitude [70]).

For the model presented here, the problem size can be expanded by increasing the complexity of the model as described in the previous
### 7.3 Improving Performance with Concurrency

#### Figure 77: Amdahl’s law for the theoretical maximum speedup when using multiple processors.

<table>
<thead>
<tr>
<th>Speedup</th>
<th>N (processors)</th>
<th>Parallel %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>95%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>90%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>50%</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>128</td>
<td></td>
</tr>
<tr>
<td></td>
<td>256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>512</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2048</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4096</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td></td>
</tr>
</tbody>
</table>

#### Figure 78: The improvement in performance as the simulation is distributed over multiple computers.

To determine the maximal speedup obtainable by distributing the simulation over multiple processors, a system of 384 nephrons was simulated for 1,000 time-steps using 1, 2, 3, 4, 6 and 12 computers (not including the computer running the controller process). The timing measurements are plotted in Fig. 78, which clearly shows that

...
the distributed implementation corresponds to $p \approx 90\%$. Accordingly, the maximal speedup that can be obtained is given by Eq. 7.7. The network capacity may have reduced the performance somewhat, and it is possible that a larger model would have reduced the frequency of communication and resulted in a larger performance increase. A larger model was not used to test this hypothesis, due to the limitations of model construction—for a description of this issue and a proposed solution, see Sect. 7.4.

$$\lim_{N \to \infty} S(N) = \frac{1}{1 - p} \approx \frac{1}{1 - 0.9} = 10$$

(7.7)

Therefore, given sufficient hardware, the distributed implementation should be capable of providing a 10-fold performance increase over the original single-threaded implementation. Based on the performance estimates for a whole-kidney simulation given in Sect. 7.2.2, this would result in the simulation of a single time-step for a whole-kidney model in 3.75 minutes (i.e., 3:45).

7.4 AUTOMATIC GENERATION OF LARGE MODELS

To investigate the dynamics of sufficiently large multi-nephron systems (e.g. whole-kidney models), it is necessary to first construct the model. The structure of the multi-nephron systems presented in Chapter 5 and Chapter 6 were specified manually, but this approach is clearly infeasible for larger systems, so there is a need for automated model growth. The kidney has a regular, but inhomogeneous, structure that cannot be easily replicated using well-known network growth techniques [4] (e.g., preferential attachment).

The automatic growth of models has been explored in the context of bone growth [29], and the dynamics of complex networks in response to the addition of nodes has been studied across many fields [4]. Such approaches are not directly applicable to the growth of multi-nephron models because the addition of a nephron to an existing model is a discrete action that cannot be performed at random, and the resulting model must have a physically-realistic structure.

Instead, two alternative approaches to generate large multi-nephron systems are proposed: a stochastic approach that uses statistical constraints on the model structure to generate a model that satisfies the constraints, and a rewriting system approach that generates the arterial tree and attaches nephrons to the tree in a stochastic manner. These two approaches are now presented in turn.

7.4.1 Stochastic generation of large models

The stochastic approach allows the modeller to specify statistical constraints on various aspects of the model structure, from which a model that satisfies these constraints is generated. Constraints can be applied at three different levels of hierarchy: nephrons, interstitial fluid columns, and the entire model. Nephrons and interstitial fluid columns can also be divided into classes, each of which has its own set of constraints.
For example, nephrons are often classified as cortical or juxta-medullary—this approach allows the modeller to define the constraints on these two classes independently. Similarly, columns of interstitial fluid can be divided into classes (e.g., based on how far into the renal pyramid they descend) and have independent sets of constraints placed on them.

The constraints on each nephron class are:

- The length and location of each tubule segment in the interstitial fluid.
- The value of nephron-specific model parameters such as $k$ (see Table 6).

The constraints on each column class are:

- The number of nephrons in the column.
- The number of branches in the column’s arterial tree.
- The distribution of nephron classes in the column.

The constraints on the entire model structure are:

- The number of interstitial fluid columns in the model.
- The number of branches in the model’s arterial tree.
- The distribution of column classes in the model.

The stochastic nature of this approach is provided by accepting three different types of value for the constraints detailed above:

- A single value (i.e., a constant);
- A range from which a value will be chosen at random (e.g., $0 \ldots 1$); and
- A list of values that associates a likelihood with each given value.

Neighbouring columns on a branch of the arterial tree are also defined to be neighbours in the 2D arrangement of columns, and the only other neighbouring columns are those from neighbouring branches. These limits on the model structure may reduce the ability of the generated models to accurately mimic a real kidney, but they are easily solved. By allowing more general constraints (such as assigning likelihoods to columns of the same class being neighbours), it can not be guaranteed that the constraints will be solvable.

7.4.2 Model growth as a rewriting system

An alternative, more complex, approach to growing such large models is to generate the arterial tree for the entire model and then attach nephrons to this tree. This requires an understanding of the renal arterial tree structure and a method for using this information to generate an arterial tree for the model. Based on experimental studies of vascular systems and the application of modelling techniques to
capture their structural properties, this approach should yield an accurate model of the renal arterial tree, to which individual nephrons can then be attached.

Vascular systems have been demonstrated to exhibit self-similarity, allowing their structure to be described by fractal models [22, 197, 163, 207]. Studies of these fractal models suggest that there is an evolutionary advantage to branching systems having a fractal dimensions [196]—the fractal models are significantly less sensitive to structural variations than the classical scaling models [163, 196, 197].

Lindenmayer systems [105, 115, 106, 107]—hereafter referred to as L-systems—are a class of rewriting system that are able to represent fractal objects obtained by use of recursive transformations [85] (see Fig. 79 for an example), such as the fractal models that have been applied to vascular systems. Indeed, such systems have already been used to generate biological models such as virtual animals [82, 83], cellular processes [64], swarms [88] and vascular systems [207] including the human bronchial tree [22] and the human retina [93].

In particular, the branching properties of the renal arterial tree [210] and the cardiovascular system [208, 209] have been shown to exhibit variations in branching parameters, resulting in multiple fractal dimensions. Radial, length and connectivity data for the arterial and venous trees of the rat kidney are also available [134].

The fractal dimension of the renal arterial tree has been estimated to be in the range 1.61–1.64 [49], which is consistent with values found in artificial fractal structures generated by diffusion-limited aggregation processes [200, 199, 198, 117] and similar to values proposed for the retinal vasculature [123, 109]. However, the processes responsible for the morphogenesis of the renal vasculature are still not fully understood [48].

A straightforward approach to building the renal arterial tree is to grammatically evolve a rewriting system so that it builds a tree with the desired fractal dimension [137] without analysing any graphical representations of the system [5]. This approach fails to take into account the multifractality of the renal arterial tree, but it is not immediately clear what an impact (if any) this would have on the model. It also uses the fractal dimension of the resulting tree as the measure of fitness, although this is not necessarily a suitable metric, as noted by Zamir in [207]:

Two such structures may have the same fractal dimension by virtue of their space-filling properties, but have widely different structures in their fluid dynamic design and function. A way of discussing fractal properties that is particularly suited to open tree structures . . . is that of considering L-system models by which the tree can be generated.

The alternative approach suggested by Zamir allows the renal arterial tree to exhibit multiple fractal dimensions, by using a parametric L-system [207] that allows the branching parameters to change as the tree descends. To determine the changes in the parameters would require dividing the renal arterial tree into a finite number of layers—each of which would ideally demonstrate little variation in its branch
(a) The rule takes a line of length L and attaches two lines to one end, with length $L \times 0.79$, at angles of $\pm 40^\circ$ to the original line.

(b) The third stage of the system.

(c) The sixth stage of the system.

(d) The tenth stage of the system.

Figure 79: An example of a Lindenmayer system that superficially resembles the arterial tree of the kidney, as presented in [207].
structure—and derive independent parameter values for each layer. The data provided by Nordsletten et al. would be a useful reference in determining how to divide the renal arterial tree into appropriate layers [134].

Both of the approaches outlined above rely on measurements of the branching characteristics of the renal arterial tree. For the purposes of growing a model of the renal arterial tree, it is enough to measure the length, diameter and branching angle of each branch in the tree [207]. Notwithstanding the techniques used in [134], an automated process built to capture these parameters would be extremely useful. Baker has demonstrated that Model-Image Registration can be used to fit a parametric 3D model to the curvature of the human cochlea, based on medical images [16]. This suggests that a similar approach may be able to determine the branching parameters based on medical images of the renal arterial tree.

If such a rewriting system were used to generate a representation of the vascular network for the model, this representation would contain sufficient information to not only generate the arterial tree of the model, but to also give each arterial branch a spatial location. Given a desired resolution for the interstitial fluid columns, their 2D arrangement would then be specified by this spatial information. As a result, this hypothetical arterial network generated by a rewriting system would specify the entire structure of the model, except for that of the individual nephrons. To specify the structure of each nephron in the model, a stochastic approach similar to the one outlined in Sect. 7.4.1 would be sufficient.
Part IV

SUMMARY OF RESEARCH
The ultimate intent of this research was to provide an approach to renal modelling that is capable of predicting whole-kidney function from the dynamics of individual nephrons, and can therefore be of practical use to clinicians. In this chapter, the contributions of this thesis are summarised. The limitations of the work are then discussed and several areas of further research to build on this work are nominated. Finally, the research outcomes are evaluated against the research aim.

8.1 CONTRIBUTIONS

The standard methods and techniques used to model nephron behaviour are not well suited to studying interactions, couplings and emergent dynamics in large multi-nephron systems. This is due to several limitations, chiefly the complexity of solving systems of coupled equations and the inability to decouple the structure and behaviour of the model. This thesis has proposed a new approach for modelling multi-nephron systems, which minimises the coupling between equations, reduces the computational cost of solving the model equations, and decouples the structure and behaviour of the model. The contributions of this thesis are:

- A modelling approach—*hierarchical dynamical networks*—which combines complex networks and graph automata into a single modelling framework. This approach explicitly captures the structure and interactions in multi-nephron systems, and decouples the structure and behaviour of the model. This approach allows emergent dynamics to be easily explored and analysed, as discussed in Chapter 4.

- The development of a multi-nephron model that produces valid behaviour in single-nephron and multi-nephron systems—as demonstrated in Chapter 5—and renders the simulation of whole-kidney function from the dynamics of individual nephrons computationally tractable. Using this model, the emergent effects of the couplings and interactions between nephrons can be investigated.

- An investigation into the dynamics of multi-nephron systems that focused on whole-system and hierarchical properties rather than the dynamics of individual nephrons, conducted in Chapter 5. As part of this investigation, the dynamics of a 72-nephron system were analysed—a system significantly larger than existing multi-nephron models.

- A study of whole-system stability in response to localised impairments in nephron function, conducted in Chapter 6. This was the first study of the emergent dynamics of impaired nephron
function, and served as an illustration of how the emergent dynamics produced by renal diseases may be predicted and analysed. The impaired multi-nephron systems exhibited very stable behaviour, which we contend is a feature of both the model and the kidney proper.

- In Chapter 7, the computational cost of the model was shown to be low enough that the simulation of whole-kidney function is feasible for the first time. It was also demonstrated that simulations could be easily distributed across multiple computers, resulting in a significant gain in performance. An implementation of the model that supports parallel and distributed execution was presented, based on the Join Calculus [59, 60].

- In order to predict whole-kidney function, a whole-kidney model must be constructed. Two approaches for automatically generating such models were proposed (but not implemented) in Chapter 7.

8.2 LIMITATIONS AND FURTHER WORK

The model presented in this thesis is built on a number of simplifying assumptions, described in Sect. 4.4.2. Most notably, all vessels are non-compliant and the function of each tubule segment was modelled at a much higher level than in existing tubule models, which attempt to capture precise low-level dynamics. However, both of these limitations can be removed by refining the update rules in the model. The update rules for each tubule node can be replaced by discrete versions of the continuous equations used in existing tubule models without incurring unexpected increases in computational cost (see Sect. 4.1.1). Similarly, compliance can be modelled by changing how the flow through a vessel is related to the pressure difference across the vessel, as illustrated in Fig. 80.

The other significant limitation on this work is the lack of a method for automatically generating models of large multi-nephron systems—without such a method, the structure of such models much be specified manually, which is infeasible for sufficiently large systems. An approach to automatically generate multi-nephron models from statistical constraints on the model structure was presented in Sect. 7.4.1, but did not provide control over the structure of the arterial tree and the 2D arrangement of the interstitial fluid columns. An improved approach was presented in Sect. 7.4.2, using a rewriting system to generate the renal arterial tree and attaching nephrons to this tree. By analysing branching characteristics in medical images of renal arterial trees, this approach would generate models with realistic arterial trees and model structure.

Several areas of further research that build on this thesis are now proposed.

8.2.1 Improved understanding of whole-kidney function

The simulation of whole-kidney function combined with the interactivity of the model would allow for an improved understanding how the
8.2 Limitations and Further Work

8.2.2 Modelling of renal disease

Based on the simple example presented in Chapter 6, detailed models of individual renal diseases can be developed and incorporated into whole-kidney models, in order to perform clinically-relevant experiments such as investigating how the presence of a particular renal disease affects whole-kidney function. The model also allows for impairments to spread across the system, which would allow for experiments into the progressive loss of kidney function to be conducted.

8.2.3 Pluggable virtual kidney

Since the modelling approach presented in Sect. 4.1 decouples the individual equations, arbitrary differential equations can be converted into difference equations and incorporated into the model. Due to the low computational cost of the model and the distributed simulation framework, a virtual kidney model could be produced that would allow researchers to experiment with incorporating systems of equations from different tubule models and to analyse the resulting dynamics at multiple levels of hierarchy, from the individual tubule to the whole-kidney scale.

Figure 80: A simple model of compliant vessels, using a cubic equation to capture the relationship between pressure and flow, rather than a linear equation. The highlighted region shows the pressure range across which the cubic equation approximates the dynamics of a compliant vessel.

whole-kidney function adapts to changes in bodily conditions. One application for this knowledge would be to design dialysis machines that can better replicate whole-kidney function.
8.2.4 Modelling kidney failure with reliability factors

A simple failure model could also be incorporated into a whole-kidney model, whereby each nephron \( n \) would be assigned a reliability factor \( R_n = f(t, v) \) that is a function of the age of the nephron \( t \) and the volume of fluid \( v \) that has flown through that nephron. Simulations of a particular model would be used to determine the specific \( R_n \) values for some \( t' \) (e.g., 60 years). These \( R_n \) values could then be applied at the start of further simulations, allowing the dynamics of an aged kidney to be analysed, since the simulation would effectively have started at time \( t = t' \). The feasibility of this study relies on choosing a reasonable function \( f(t, v) \)—given any such \( f(t, v) \), analysis of the model dynamics and disease modelling experiments should prove to be useful benchmarks to compare the resulting dynamics against those observed in patients of the particular age \( t' \).

8.3 Conclusion

The ultimate intent of this research was to provide the basis for an approach to renal modelling that is capable of predicting whole-kidney function from the dynamics of individual nephrons, and can therefore be of practical use to clinicians.

A unique modelling approach was proposed in Chapter 4, and the resulting models were shown to produce valid behaviour in Chapter 5. Chapter 5 also demonstrated that the models could be used to study emergent dynamics in multi-nephron systems, and the application of this analysis to studying the effects of localised impairments on whole-system dynamics was illustrated in Chapter 6. Finally, the computational cost of the model was investigated in Chapter 7, which demonstrated that the simulation of whole-kidney function was computationally tractable, and that the performance of the simulations could be improved by utilising multiple computers.

In conclusion, the modelling and analysis techniques presented in this thesis allow for emergent dynamics to be studied in large multi-nephron systems. This work demonstrates that, for the first time, simulation of whole-kidney function from the dynamics of individual nephrons is tractable. Furthermore, the work provides a basis for predicting emergent effects of localised renal disease. With the continued development of this model, we hope that significant insight will be gained into the onset, progression and treatment of renal diseases.
BIBLIOGRAPHY


COLOPHON

This thesis was typset with \LaTeX{} using the Bera, Euler and Times type faces. The typographic style is available for \LaTeX{} via CTAN as "classicthesis", except that this thesis uses Bera as the main font and Times for smallcaps.

Final Version as of 1st September 2008 at 20:59.
Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
MOSS, R.

Title:
A clockwork kidney: using hierarchical dynamical networks to model emergent dynamics in the kidney

Date:
2008

Citation:

Publication Status:
Unpublished

Persistent Link:
http://hdl.handle.net/11343/35092

File Description:
A clockwork kidney - print version

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.